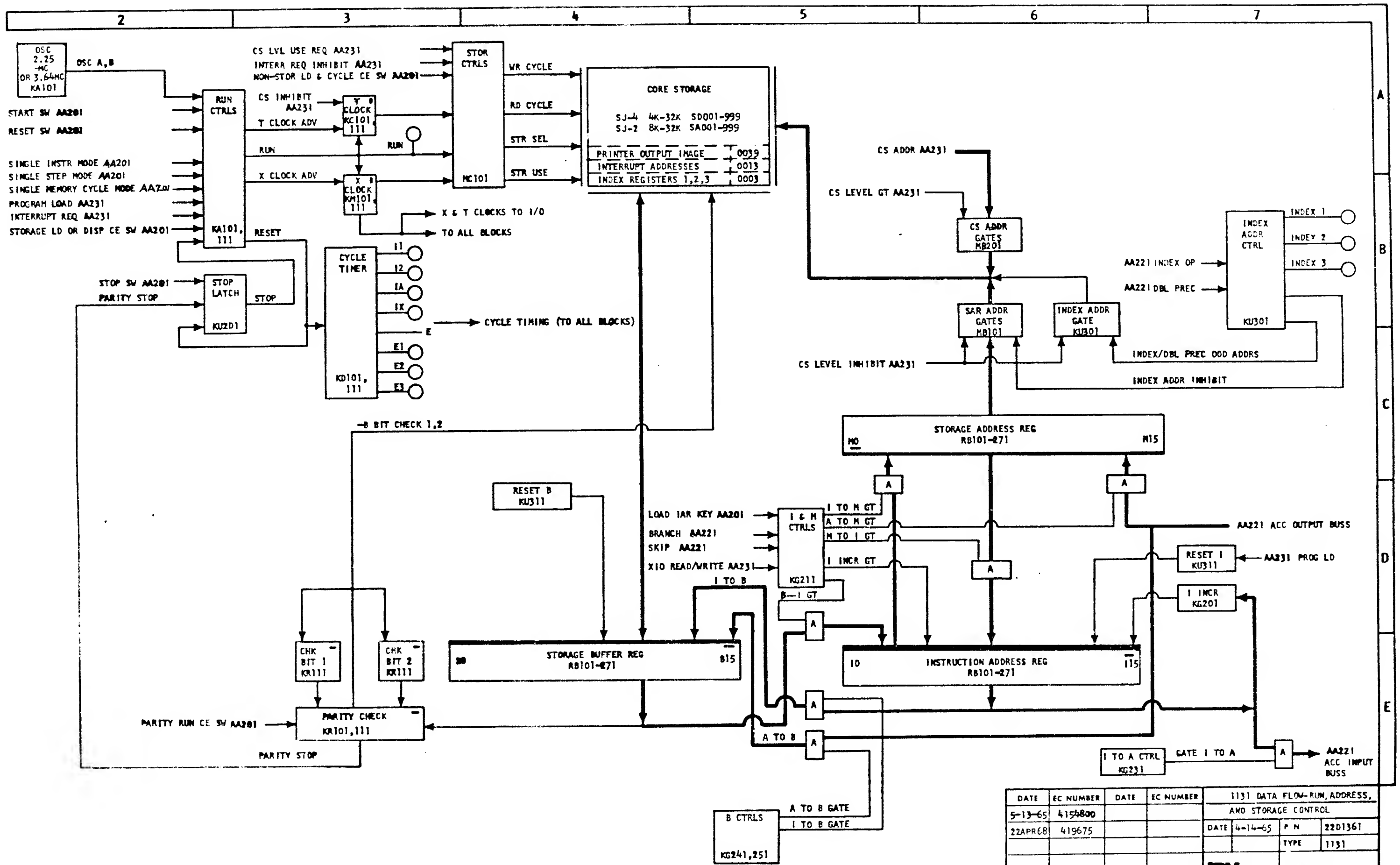
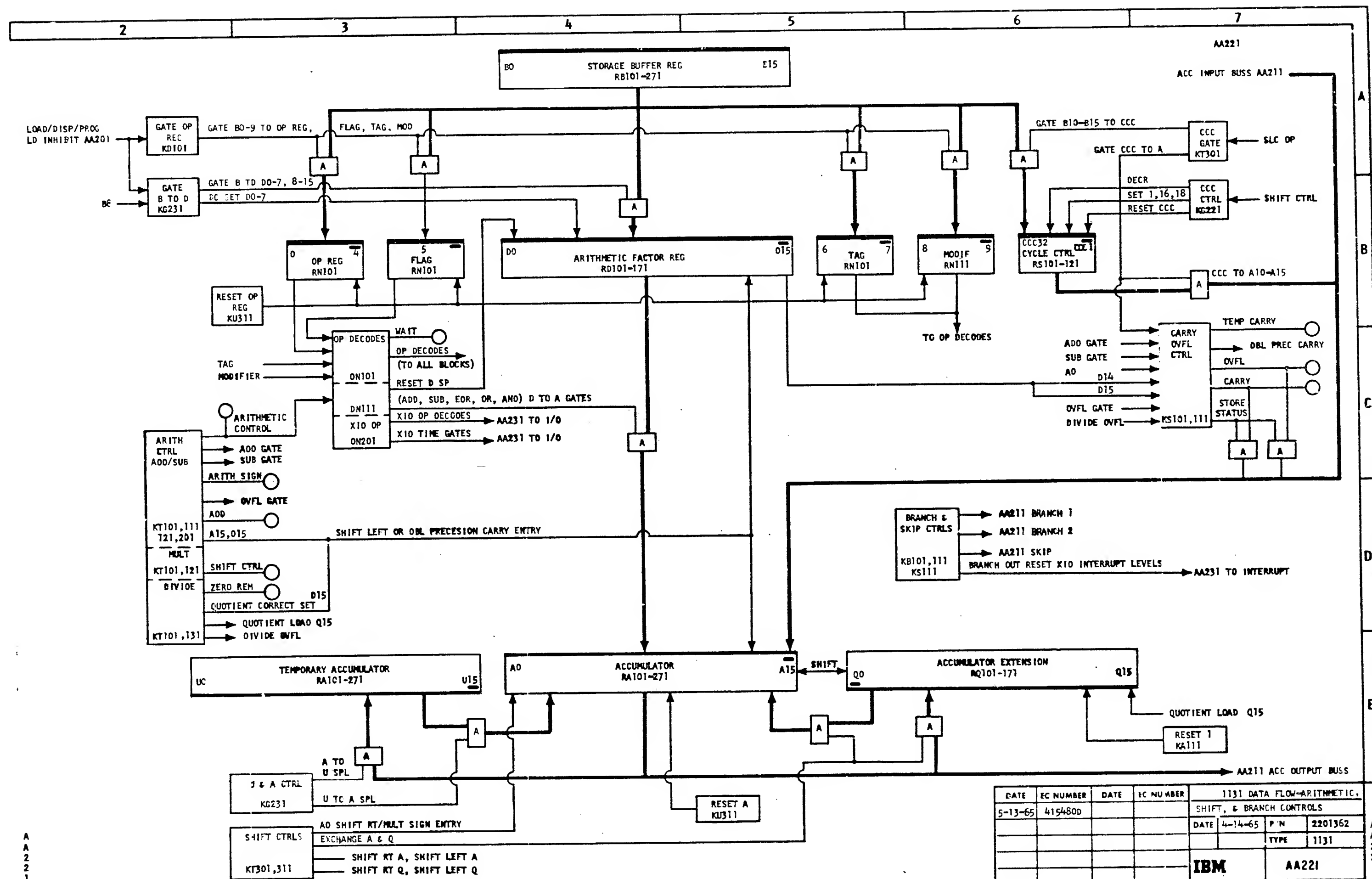
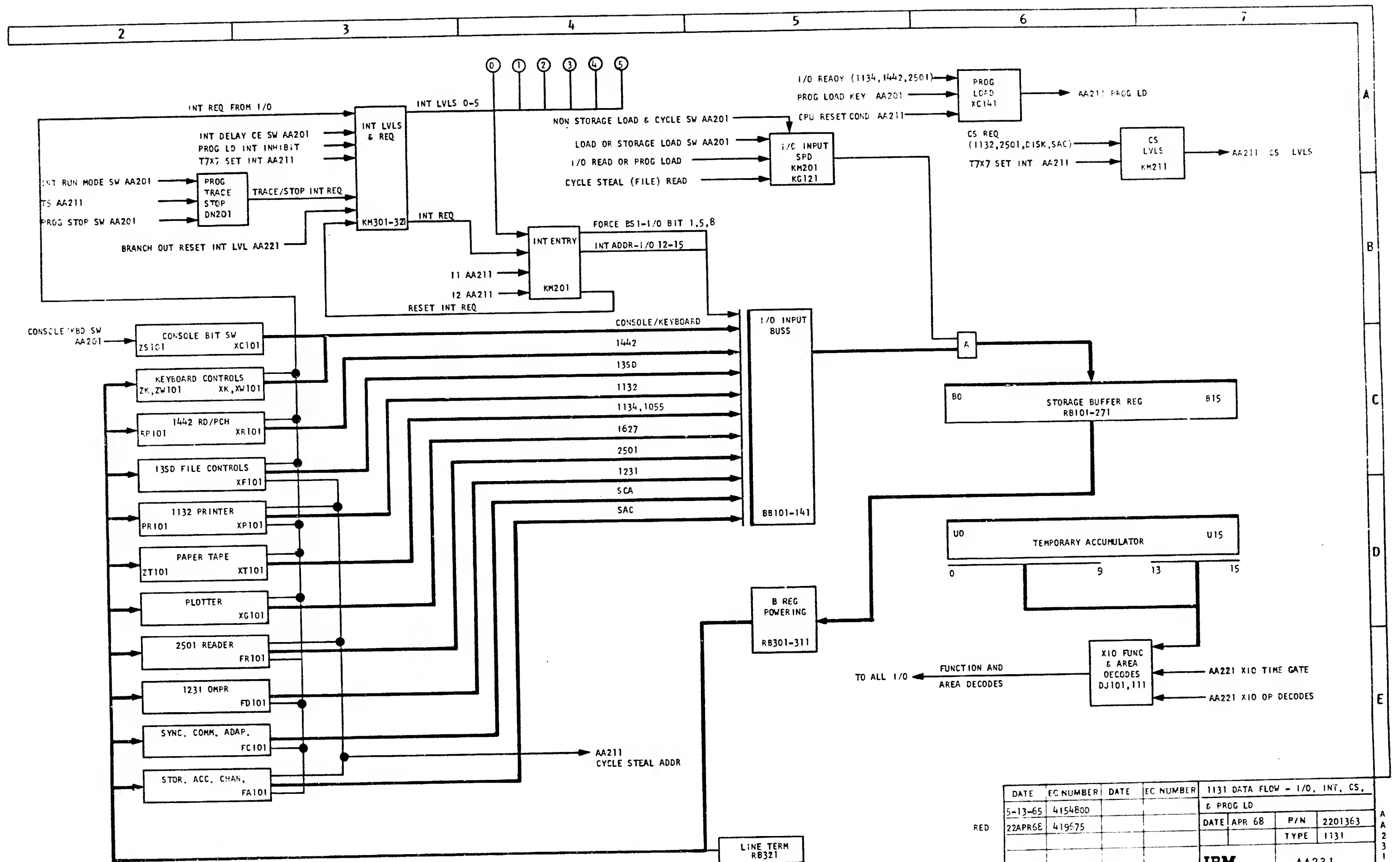


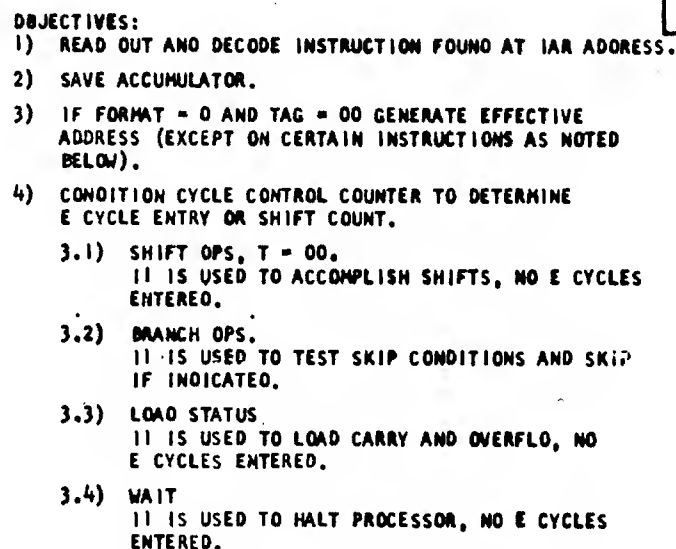
1130 SYSTEM DATA FLOW			
DATE	EC NUMBER	DATE	EC NUMBER
5-13-65	4154803	4-19-65	#/N
OCT 65	415483A		TYPE
22APR68	419675		1131
		IBM	
		AA/01	



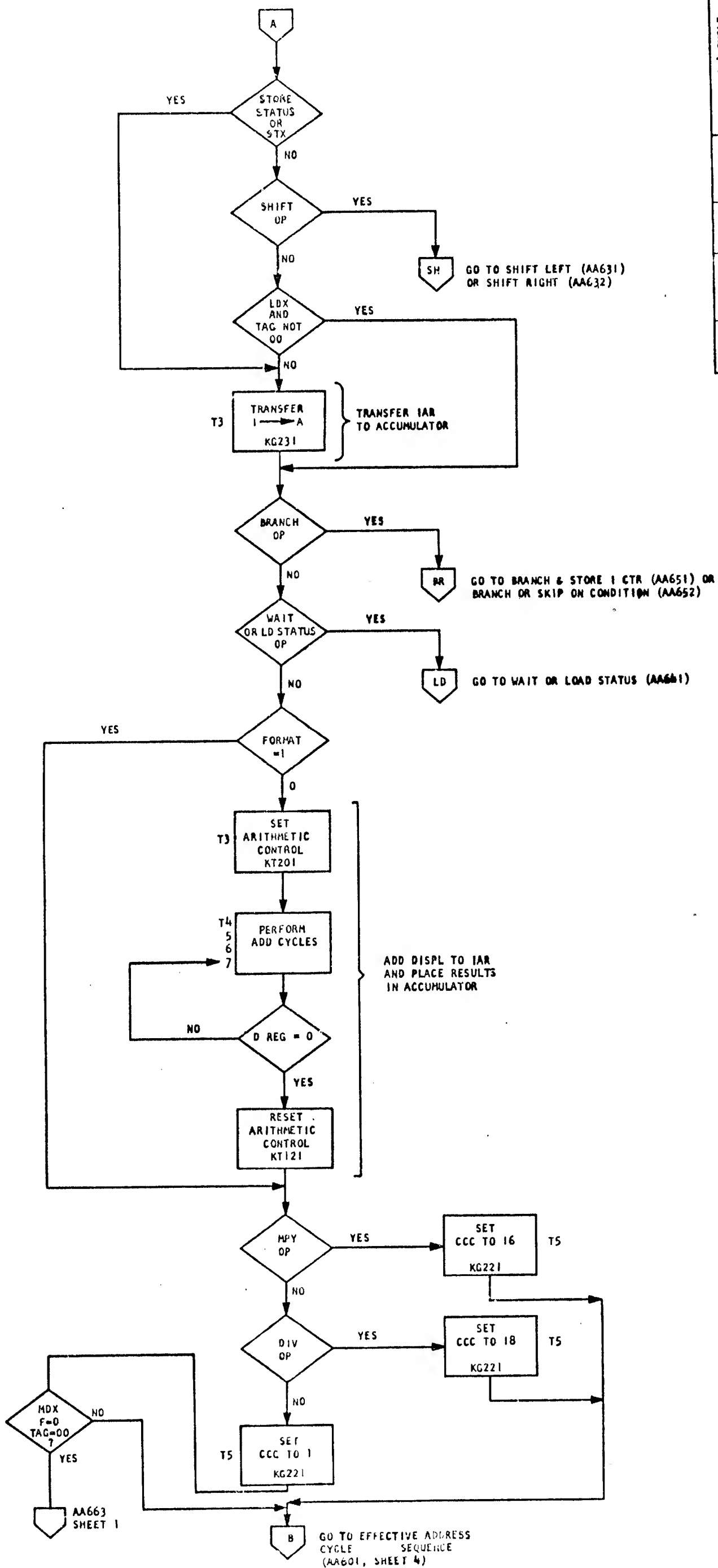
DATE	EC NUMBER	DATE	EC NUMBER	1131 DATA FLOW-RUN, ADDRESS, AND STORAGE CONTROL			
5-13-65	4154800			DATE	4-14-65	P N	22D1361
22APR68	419675					TYPE	1131
				IBM		AA211	



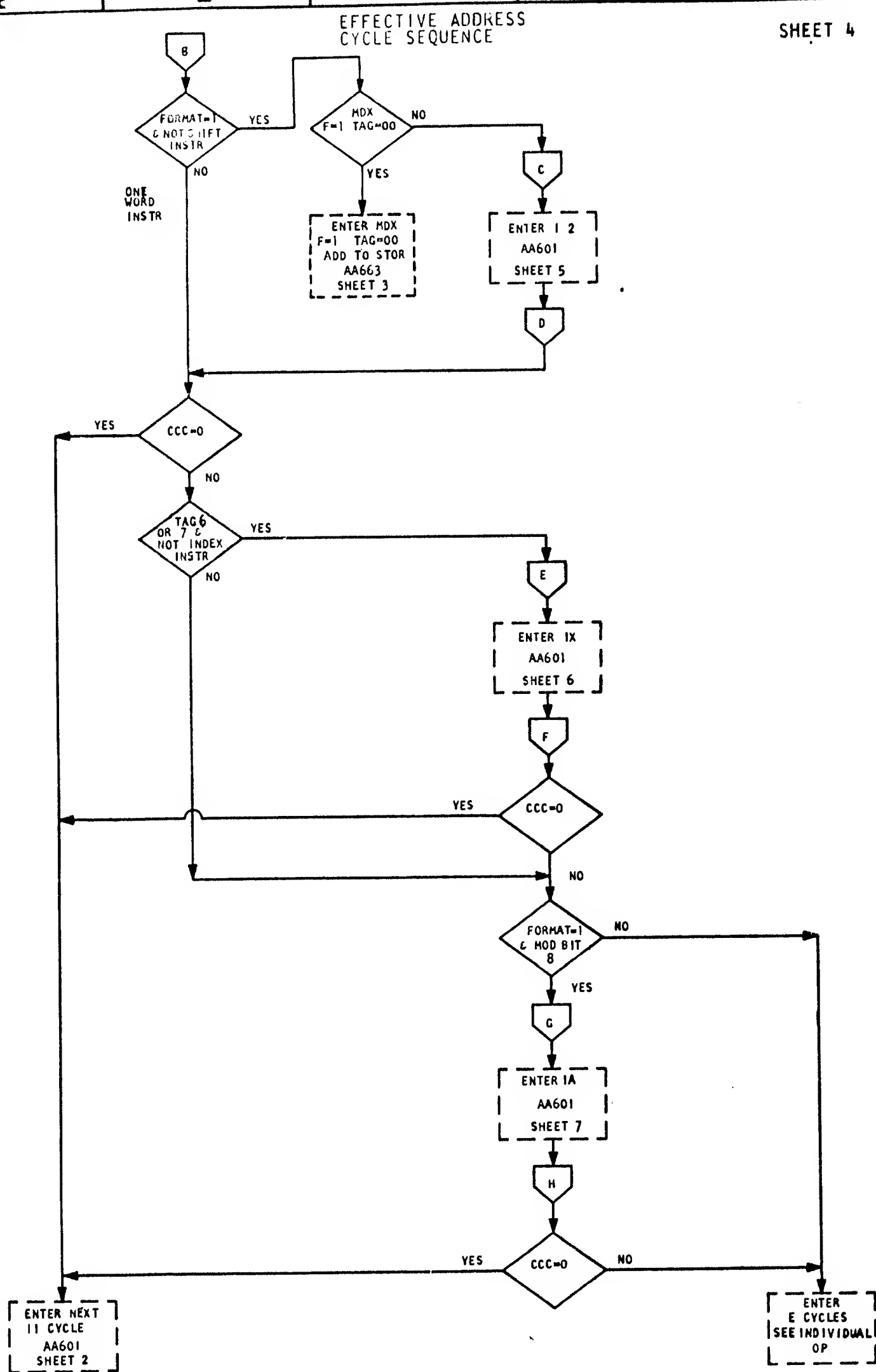




DATE	EC NUMBER	DATE	EC NUMBER	I-1 CYCLE
	415480D			
OCT 65	415483A			
MAY 67	419633			DATE 5-5-65 P/N 2204H25
22APR68	419675			TYPE 1131
				IBM AA601

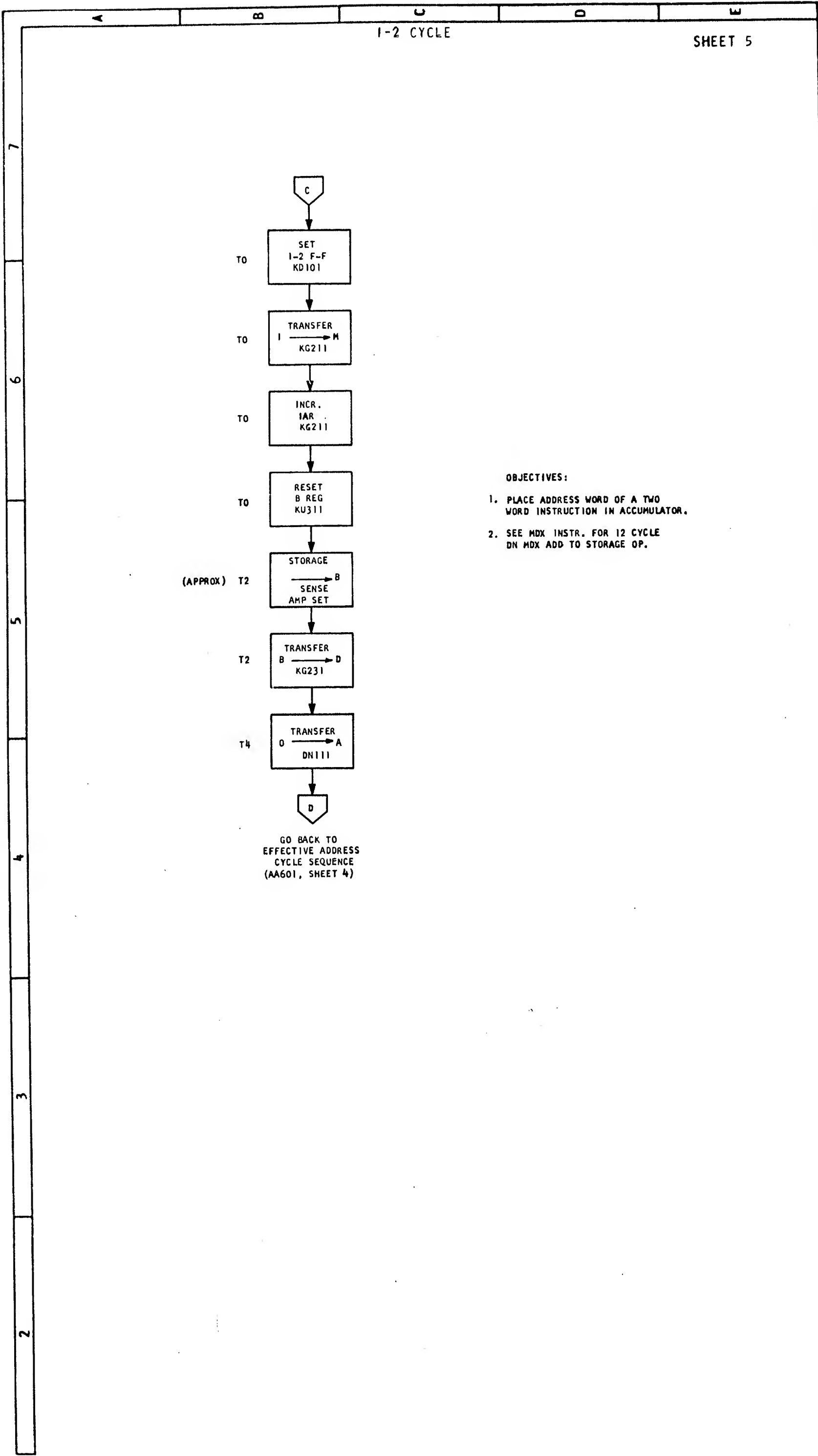


I-1 CYCLE		EC NUMBER		DATE		P N		TYPE		AA601	
DATE	415480D	EC NUMBER	415483A	DATE	5-5-65	P N	2201425	TYPE	1131	IBM	AA601
	OCT 65		MAY 67								
	22APR68		419675								



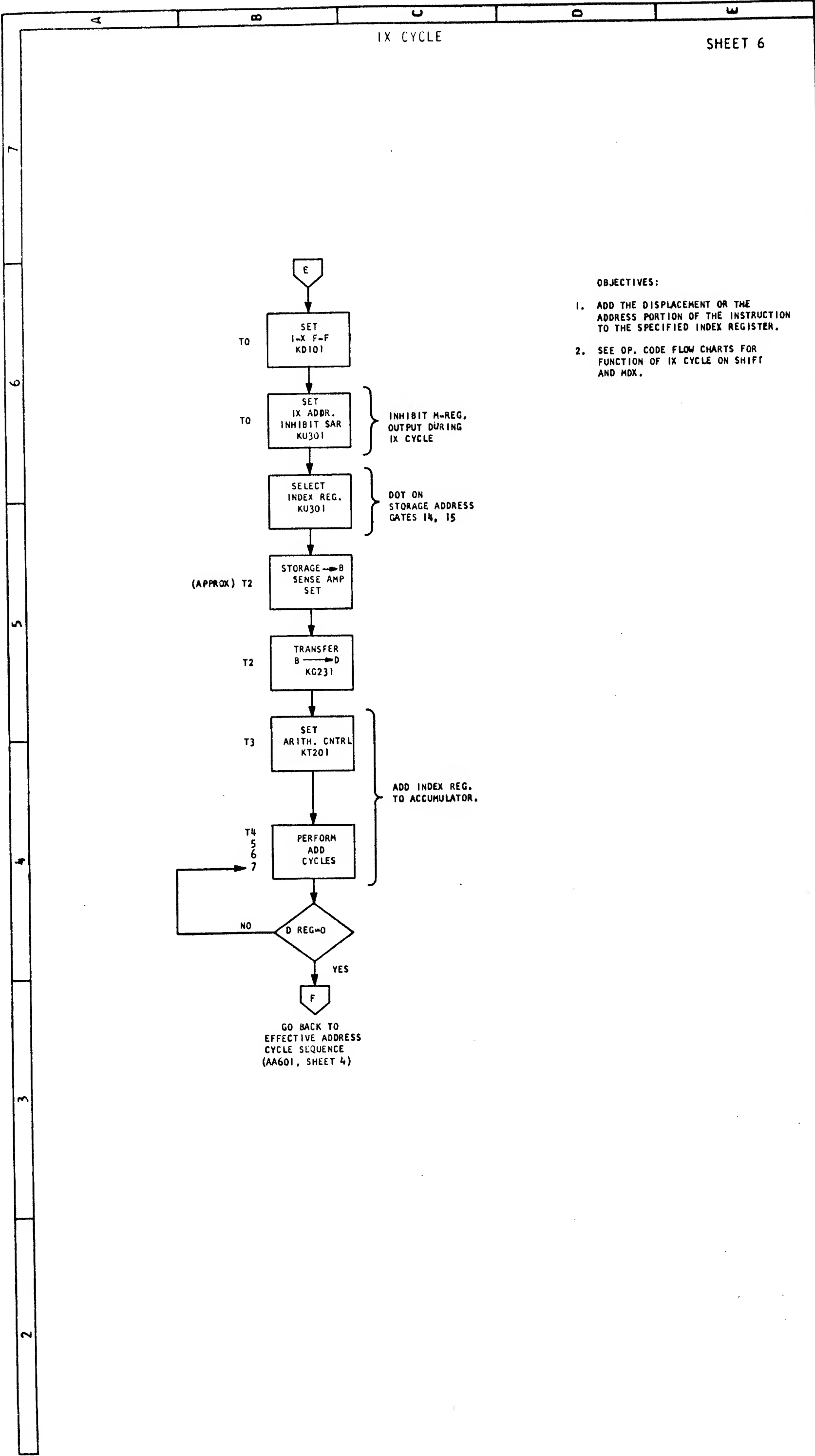
EFFECTIVE ADDRESSING SEQUENCE							
FORMAT	TAG	MOD 8	11	12*	1X**	1A	EFFECTIVE ADDRESS
0	00	-	/				DISPLACEMENT + IAR + 1
0	NOT 00	-	/		/		DISPLACEMENT & INDEX REG CONTENTS
1	00	0	/	/			ADDRESS
1	00	1	/	/		/	(ADDRESS) CONTENTS
1	NOT 00	0	/	/	/		ADDRESS & INDEX REG CONTENTS
1	NOT 00	1	/	/	/	/	(ADDRESS & INDEX REG CONT) CONTENTS

* NOT SHIFT INSTR. ** NOT INDEX INSTR.



- OBJECTIVES:
1. PLACE ADDRESS WORD OF A TWO WORD INSTRUCTION IN ACCUMULATOR.
 2. SEE MDX INSTR. FOR 12 CYCLE DN MDX ADD TO STORAGE OP.

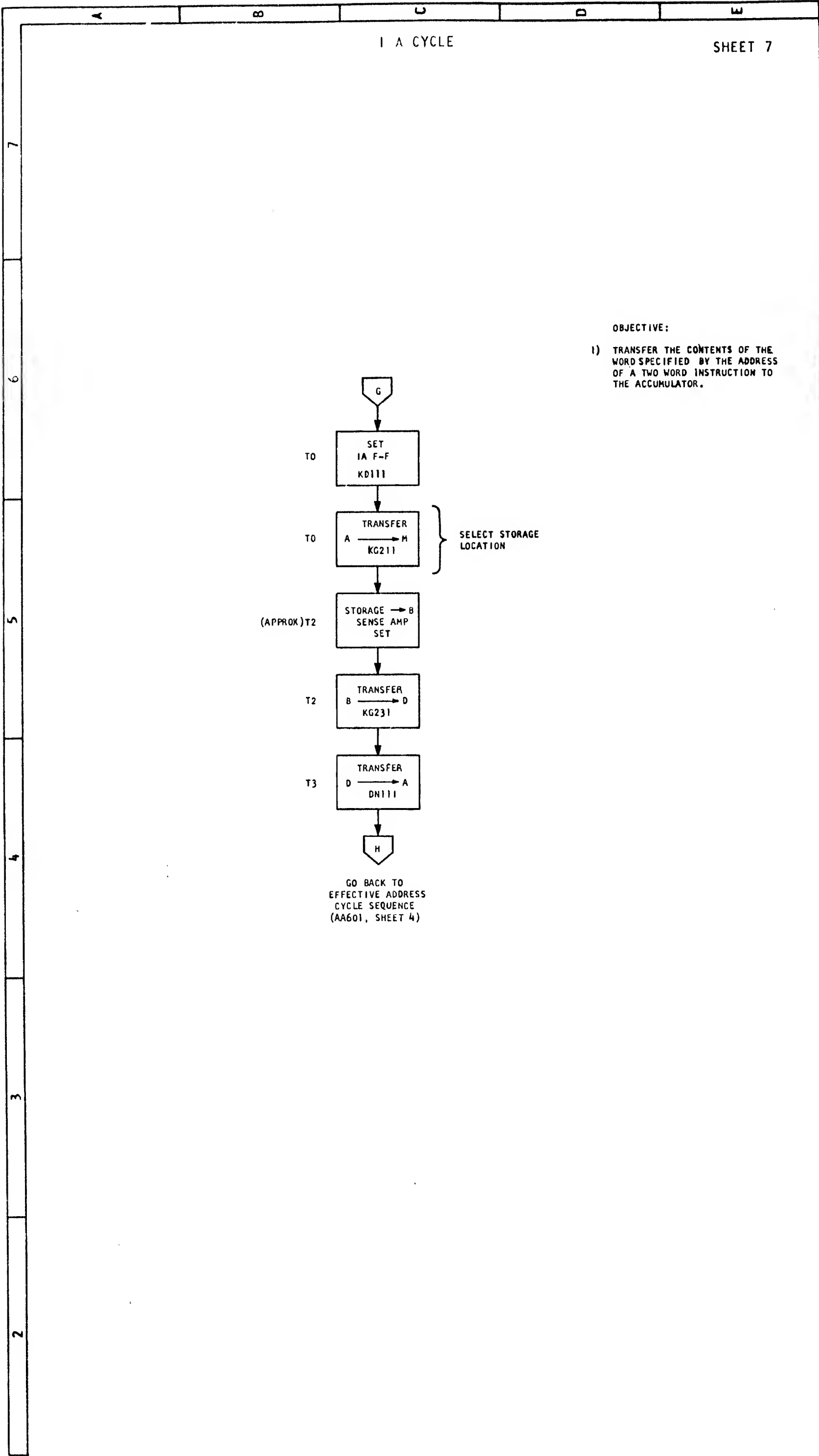
1-2 CYCLE		1-2 CYCLE	
DATE	EC NUMBER	DATE	EC NUMBER
	4154800		
OCT 65	415483A		
MAY 67	419633		
22APR68	419675		
IBM		AA601	



OBJECTIVES:

1. ADD THE DISPLACEMENT OR THE ADDRESS PORTION OF THE INSTRUCTION TO THE SPECIFIED INDEX REGISTER.
2. SEE OP. CODE FLOW CHARTS FOR FUNCTION OF IX CYCLE ON SHIFT AND MDX.

IX CYCLE		EC NUMBER		DATE	
DATE	EC NUMBER	415480D			
OCT 65	415483A				
MAY 67	419633				
22APR68	419675				
IBM		AA601			



IA CYCLE		EC NUMBER		DATE	
DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER
OCT 65	415480D			5-24-55	2201425
MAY 67	415483A				P/N
22APR68	419675				TYPE
					1131
					AA601

INTERRUPT
FORCED BRANCH AND STORE I CTR INDIRECT

SHEET 1

OBJECTIVE:

TO SERVICE AN INTERRUPT REQUEST BY FORCING A BSI INSTRUCTION WHICH WILL: a. BRANCH TO THE INTERRUPT SUBROUTINE FOR THE APPROPRIATE LEVEL. b. STORE THE I CTR TO PERMIT A RETURN TO THE MAIN LINE PROGRAM.

REQ LEVEL	11	12	13	14	15
0		1			
1		1			1
2		1		1	
3		1		1	1
4		1	1		
5		1	1		1
6		1	1	1	
7		1	1	1	1

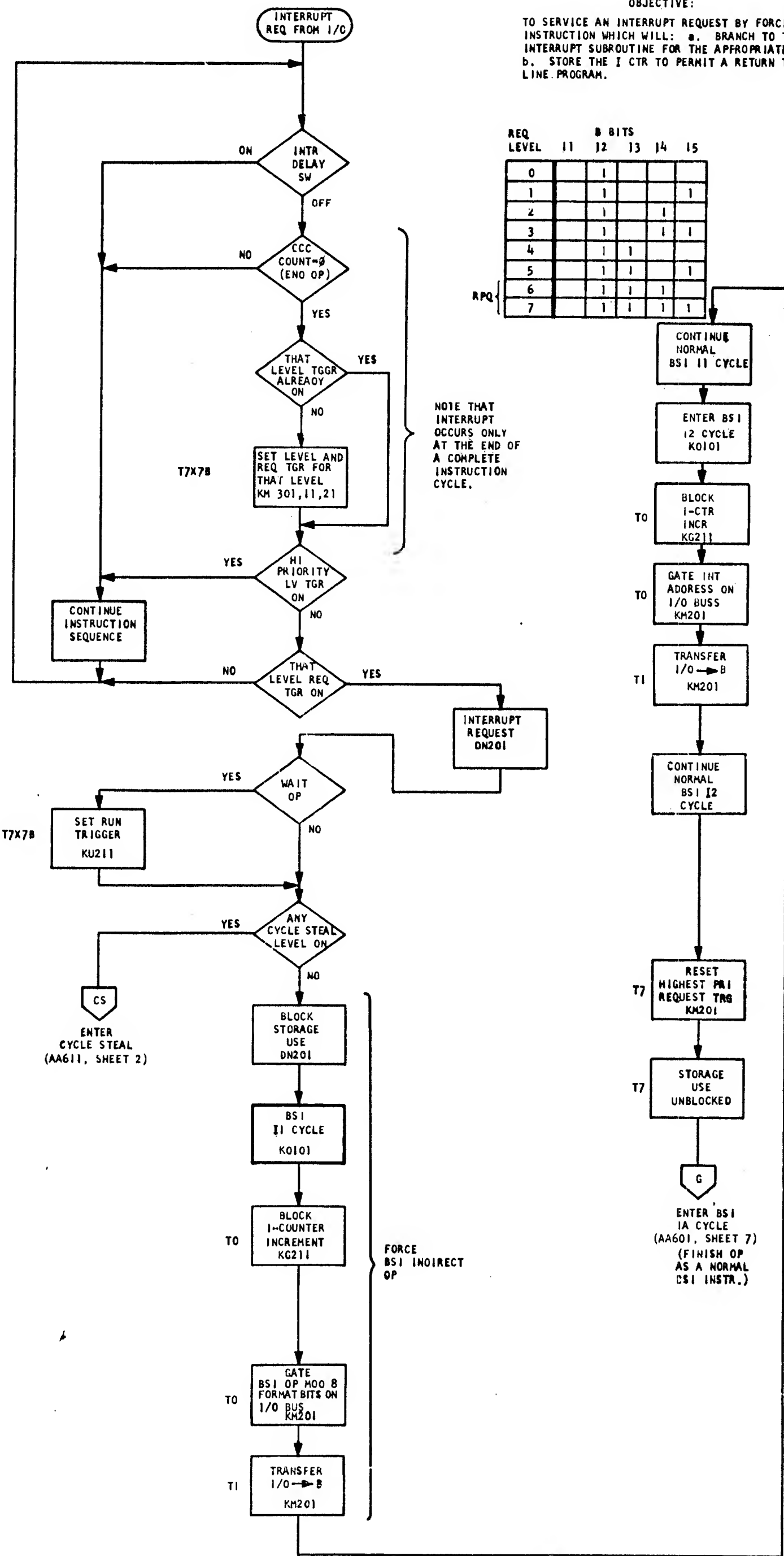
RPQ

NOTE THAT INTERRUPT OCCURS ONLY AT THE END OF A COMPLETE INSTRUCTION CYCLE.

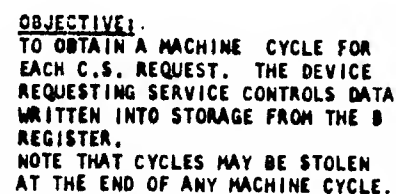
FORCE BSI INDIRECT ADDRESS TO INTERRUPT SUBROUTINE

FORCE BSI INDIRECT OP

ENTER BSI 1A CYCLE (AA601, SHEET 7) (FINISH OP AS A NORMAL CSI INSTR.)



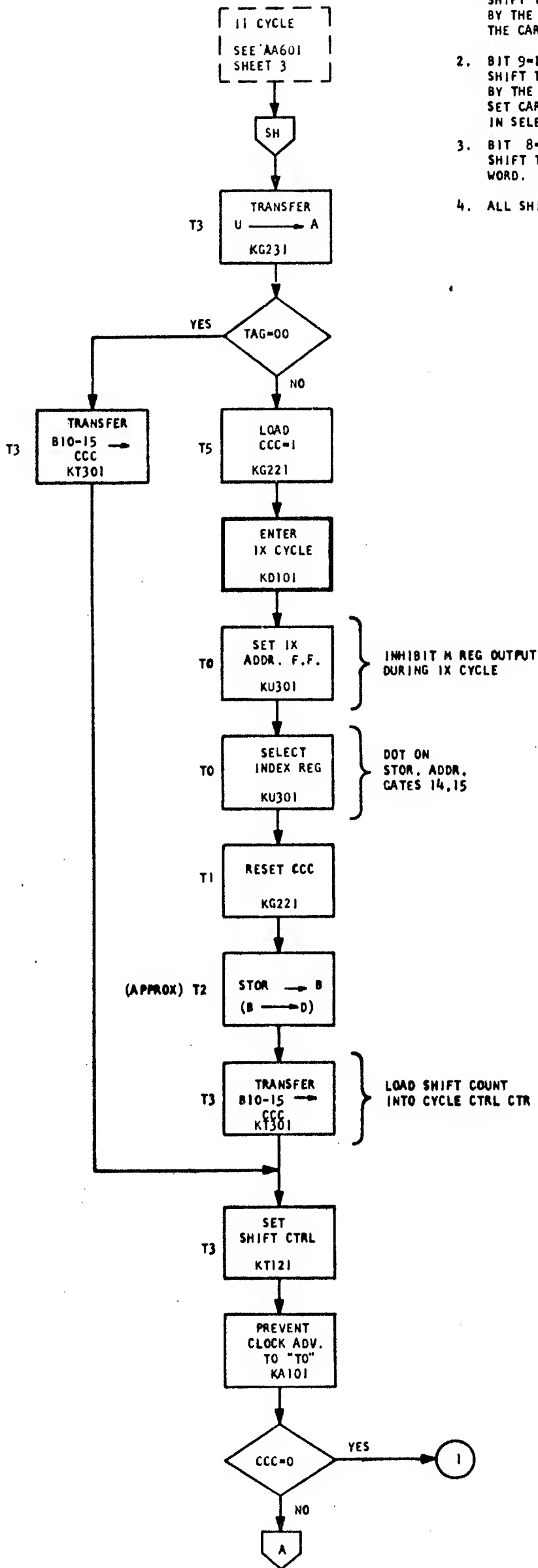
INTERRUPT FORCED BRANCH		AND STORE I CTR INDIRECT		P/N		TYPE		IBM	AA611
DATE	EC NUMBER	DATE	EC NUMBER	DATE	P/N	DATE	TYPE		
	415480D				2201432		1131		
OCT 65	415483A								



DATE	EC NUMBER	DATE	IC NUMBER	CYCLE STEAL
	415480D			
OCT 65	415483A			DATE 3-25-65 P/M 2201432
				TYPE 1131
				IBM AA611

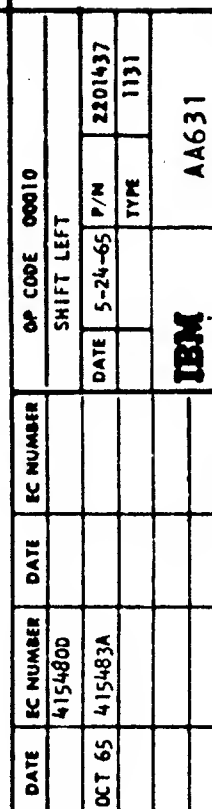
OBJECTIVES:

- 1. BIT 8=0 (SLA OP)
SHIFT THE ACCUM LEFT THE NO. OF POSITIONS INDICATED BY THE SHIFT CTR (CCC). BIT LEAVING A0 SHIFTS INTO THE CARRY TRG.
- 2. BIT 9=1, TAG #00 (SLC OP)
SHIFT THE ACCUM LEFT THE NO. OF POSITIONS INDICATED BY THE CCC. TERMINATE SHIFT IF ONE IS FOUND IN A0, SET CARRY TRG ON, AND PLACE REMAINDER OF CCC COUNT IN SELECTED INDEX REG.
- 3. BIT 8=1
SHIFT THE ACCUM AND THE Q REG AS A 32 BIT DOUBLE PRECISION WORD.
- 4. ALL SHIFT OPS ARE SHORT FORMAT (F=0) ONLY.



TAG	CCC SHIFT CNT
00	DISPLACEMENT
01	INDEX 1
10	INDEX 2
11	INDEX 3

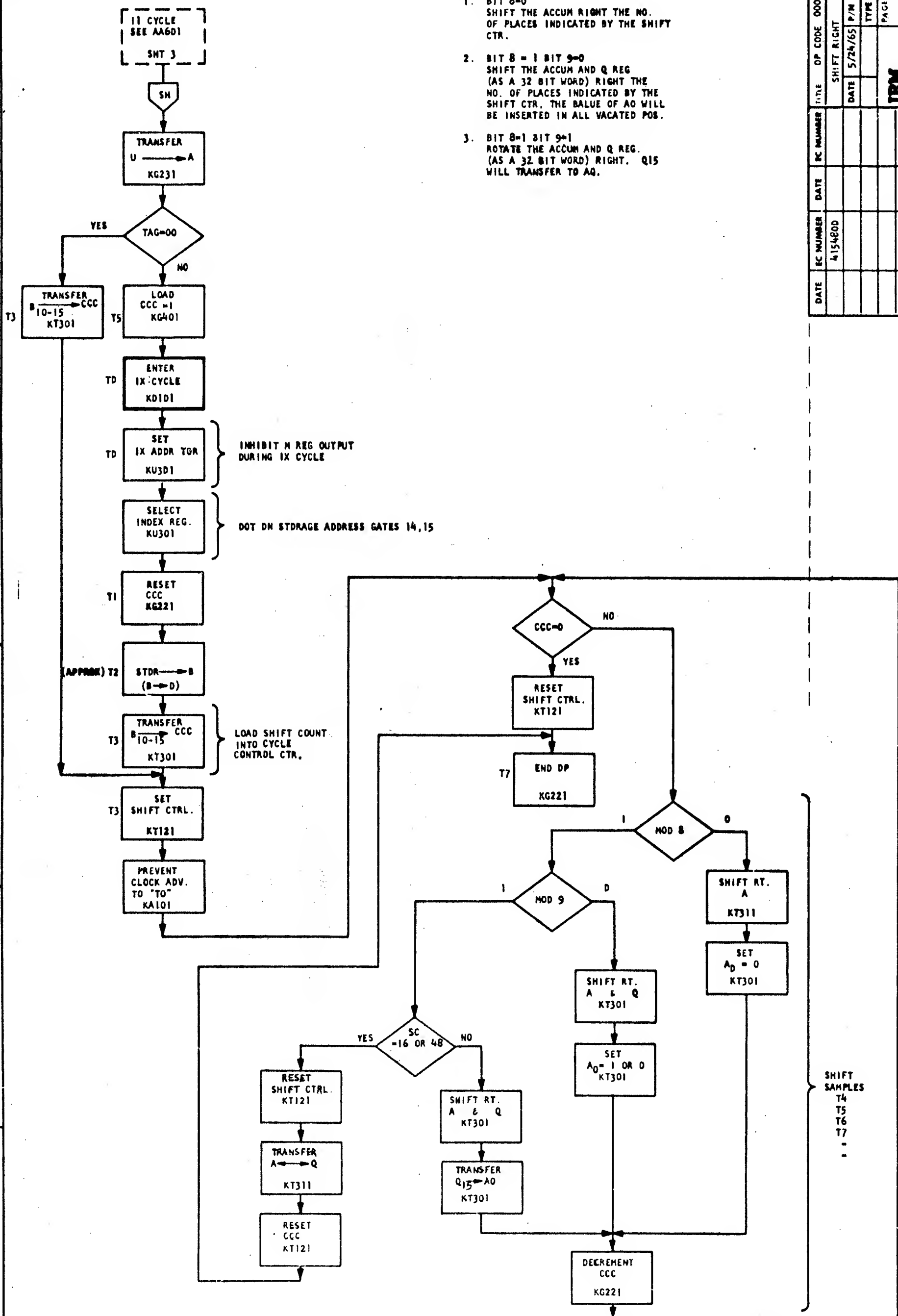
OP CODE 00010		SHIFT LEFT		P/N	TYPE	AA631
DATE	EC NUMBER	DATE	EC NUMBER			
OCT 65	4154800			2201437	1131	
	415483A					



SHIFT
SAMPLES
T4
T5
T6
T7
-
-

OBJECTIVES:

1. BIT 8-0
SHIFT THE ACCUM RIGHT THE NO.
OF PLACES INDICATED BY THE SHIFT
CTR.
2. BIT 8 = 1 BIT 9-0
SHIFT THE ACCUM AND Q REG
(AS A 32 BIT WORD) RIGHT THE
NO. OF PLACES INDICATED BY THE
SHIFT CTR, THE VALUE OF AQ WILL
BE INSERTED IN ALL VACATED POS.
3. BIT 8-1 BIT 9-1
ROTATE THE ACCUM AND Q REG.
(AS A 32 BIT WORD) RIGHT. Q15
WILL TRANSFER TO AQ.



SHIFT
SAMPLES
T4
T5
T6
T7
-
-

OP CODE 00100 LOAD STATUS

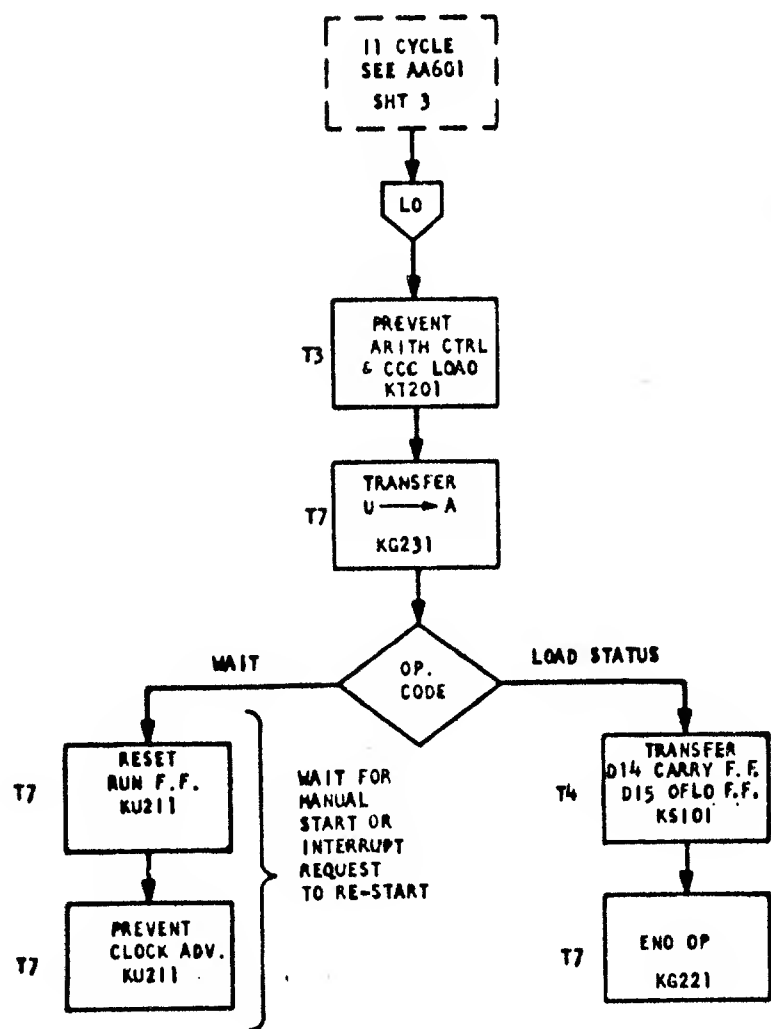
OP CODE { 00000
00110
00111
01010
01011
01111
10110
10111
11111 } WAIT

LOAD STATUS
OBJECTIVES:

- 1. LOAD THE CARRY INDICATOR WITH THE STATUS OF BIT 14 IN THE INSTRUCTION WORD.
- 2. LOAD THE OVERFLOW INDICATOR WITH THE STATUS OF BIT 15 IN THE INSTRUCTION WORD.
- 3. A "1" IN THESE BIT POSITIONS WILL TURN ON THE INDICATOR.
- 4. A "0" IN THESE BIT POSITIONS WILL TURN OFF THE INDICATOR.

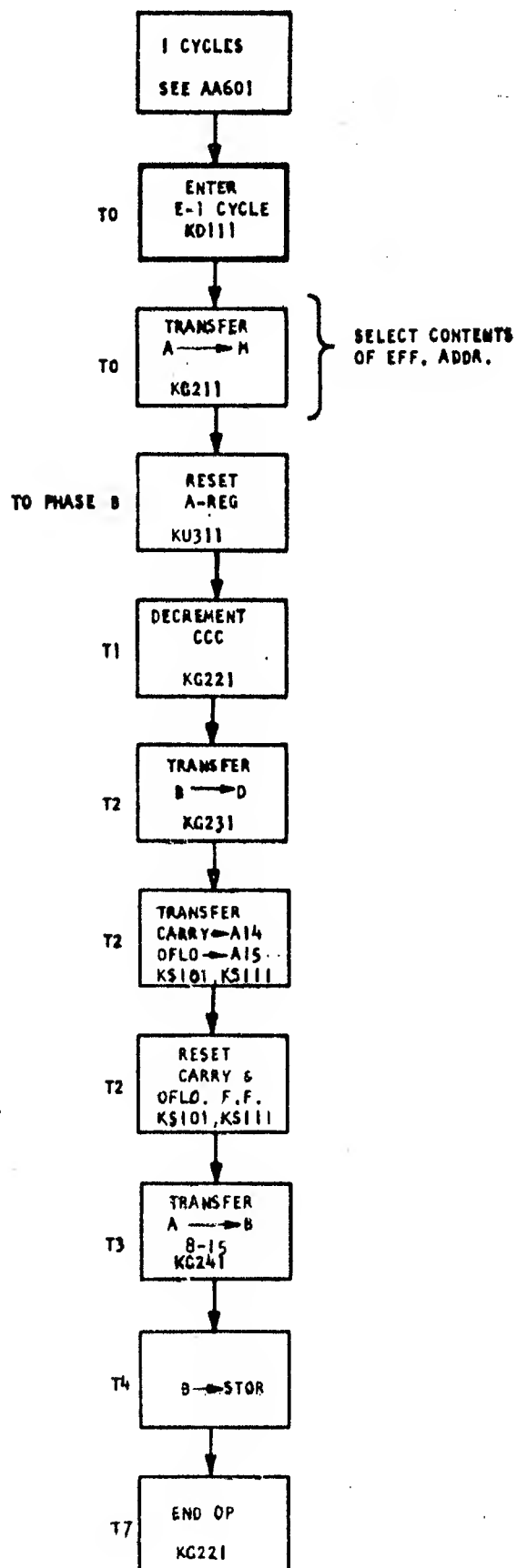
WAIT
OBJECTIVES:

- 1. STOP THE PROCESSOR IN A WAIT CONDITION
- 2. LOAD STATUS AND WAIT ARE VALID IN SHORT FORMAT (F=0) ONLY.



OP CODE 00100 LOAD STATUS			
OP CODE		WAIT	
DATE	P/N	DATE	P/N
5-24-65	2201440	1131	AA641
TYPE		IBM	
RC NUMBER		AA641	
DATE		AA641	

1. STORE THE STATUS OF THE CARRY INDICATOR INTO BIT 14 OF THE CONTENTS OF THE EFF. ADDR.
2. STORE THE STATUS OF THE OVERFLOW INDICATOR INTO BIT 15 OF THE CONTENTS OF THE EFF. ADDR.
3. IF THE INDICATOR IS ON A "ONE" WILL BE STORED.
4. IF THE INDICATOR IS OFF A "ZERO" WILL BE STORED.
5. RESET CARRY AND OVERFLOW.

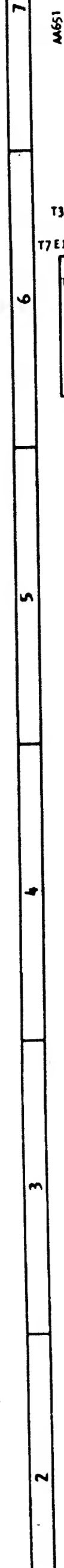


DATE	EC NUMBER	DATE	EC NUMBER	OP CODE	00101
	415480D			STORE STATUS	
				DATE	5-24-65
				P/M	2201441
				TYPE	1131
				IBM	
				AA642	

2	3	4	5	6	7
---	---	---	---	---	---

T3
T7E

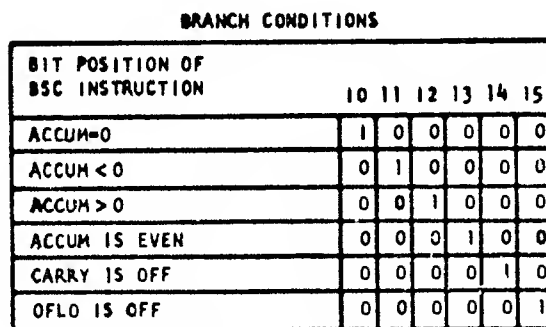
AA651



BIT POSITION OF BS1 INSTRUCTION		10	11	12	13	14	15
ACCUM = 0		1	0	0	0	0	0
ACCUM < 0		0	1	0	0	0	0
ACCUM > 0		0	0	1	0	0	0
ACCUM IS EVEN		0	0	0	1	0	0
CARRY IS OFF		0	0	0	0	1	0
OFLO IS OFF		0	0	0	0	0	1

1. F=0 STORE THE I-CTR. IN THE EFF. ADDR. THE NEXT INSTRUCTION PERFORMED WILL BE AT E.A.+1
2. F=1 IF NONE OF THE BRANCH CONDITIONS BEING TESTED IS TRUE, THE I-CTR. IS STORED AT THE EFF. ADDR. THE NEXT INSTR. PERFORMED WILL BE IN EA+1.
3. F=1 IF ANY OF THE BRANCH CONDITIONS BEING TESTED ARE TRUE, THE NEXT INSTRUCTION IN SEQUENCE IS PERFORMED.
4. RESET OVERFLO IF TESTED

4455-



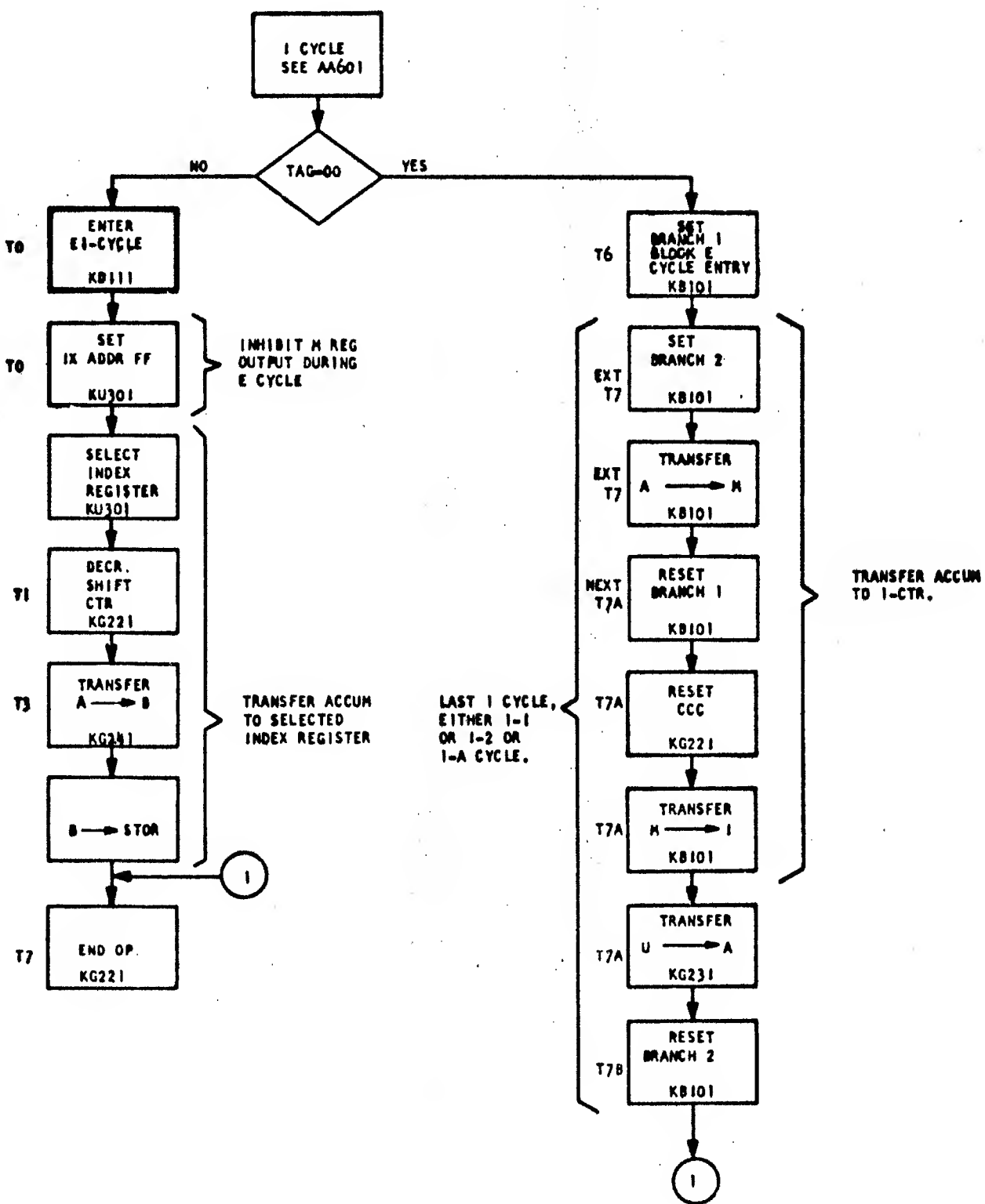
- OBJECTIVES:**
1. $F=0$ IF ANY OF THE CONDITIONS BEING TESTED ARE TRUE, THE I-CTR WILL SKIP THE NEXT ONE WORD INSTR.
 2. $F=1$ IF NONE OF THE CONDITIONS BEING TESTED IS TRUE THE PROGRAM WILL BRANCH TO THE EFF. ADDR.
 3. MOD 9=1 IF INSTR. SKIPS WITH $F_5=0$ OR DOES NOT SKIP WITH $F_5=1$ (BRANCH) RESET HIGHEST PRIORITY INTERRUPT LEVEL IN OPERATION.
 4. RESET OVERFLO IF TESTED.

PLACE BRANCH
EFFECTIVE ADDRESS
IN IAR.

OP CODE 01100 LOAD INDEX

- OBJECTIVES:
- 1. F=0 LOAD THE I CTR. OR AN INDEX REGISTER WITH THE DISPL.
 - 2. F=1 LOAD THE I CTR. OR AN INDEX REGISTER WITH THE ADDR. PORTION OF THE INSTRUCTION.
 - 3. F=1 MOD 8 LOAD THE I CTR OR AN INDEX REGISTER WITH THE CONTENTS OF THE STORAGE LOCATION SPECIFIED BY THE ADDRESS.

TAG	LOAD
00	I CTR
01	INDEX 1
10	INDEX 2
11	INDEX 3



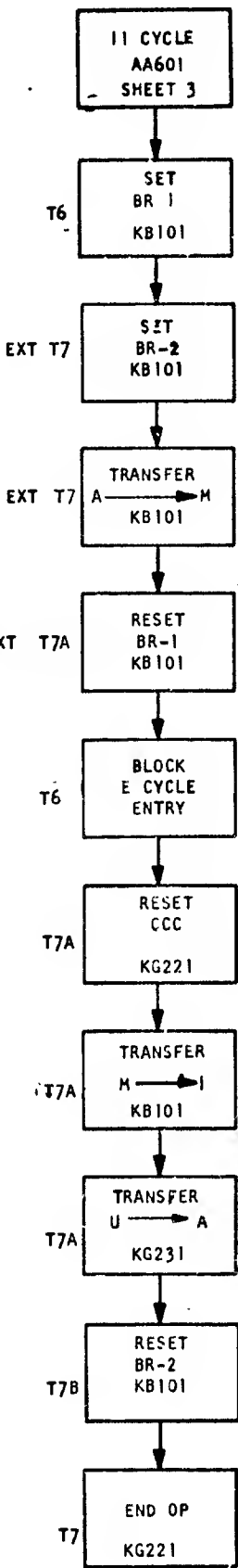
DATE	IC NUMBER	DATE	IC NUMBER	TITLE	OF CODE 01100
	4154808			LOAD INDEX	
				DATE	5-24-65
				P/N	2201446
				TYPE	1131
				PAGE NO	AA661



IBM

OP CODE 0110 MODIFY INDEX
AND SKIP FORMAT = 0 TAG = 00

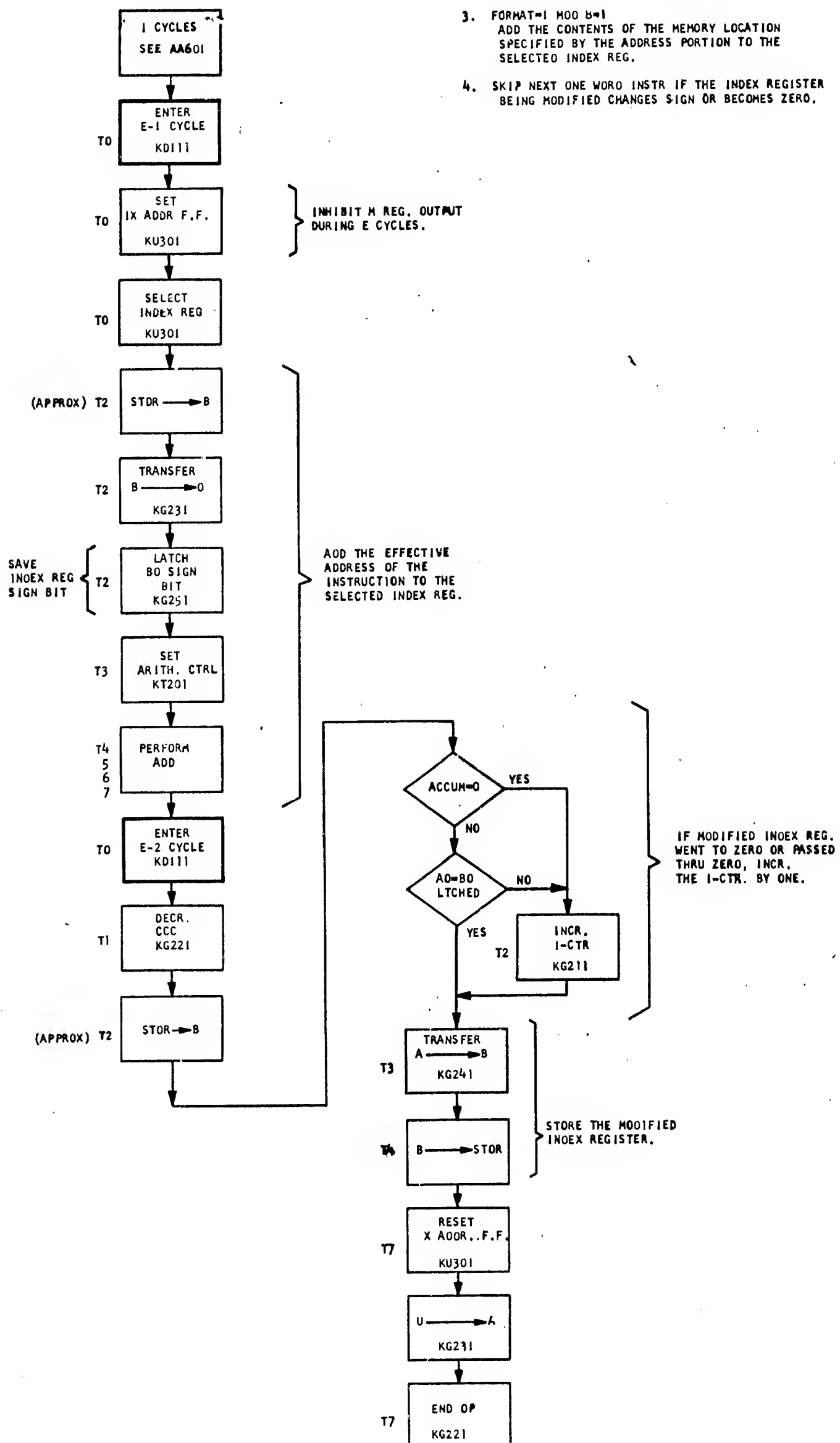
SHEET 1



OBJECTIVE:
ADD THE DISPLACEMENT TO THE I-COUNTER.
THE NEXT INSTRUCTION WILL BE LOCATED AT
(DISPL.) + (I-CTR.+1). THIS PROVIDES AN
EFFECTIVE BRANCH TO THE NEW IAR VALUE.

OP CODE 0110 MODIFY INDEX			
AND SKIP FORMAT=0 TAG=00			
DATE	EC NUMBER	DATE	EC NUMBER
4154800		5-24-65	
OCT 65	415483A	P/N	2201448
JAN 66	415726	TYPE	1131
22AFR65	419675		
		IBM	
		AA663	

1. OBJECTIVES:
FORMAT=0
ADD DISPLACEMENT TO SELECTED INDEX REGISTER.
2. FORMAT=1 MOO 8=0
ADD THE ADDRESS PORTION OF THE INSTRUCTION TO THE SELECTED INDEX REGISTER.
3. FORMAT=1 MOO 8=1
ADD THE CONTENTS OF THE MEMORY LOCATION SPECIFIED BY THE ADDRESS PORTION TO THE SELECTED INDEX REG.
4. SKIP NEXT ONE WORD INSTR IF THE INDEX REGISTER BEING MODIFIED CHANGES SIGN OR BECOMES ZERO.



DATE	EC NUMBER	DATE	EC NUMBER	OP CODE 01110 MODIFY
	415480D			INDEX & SKIP TAG#00
OCT 65	415483A			DATE 5-24-65 P/N 2201448
JAN 66	415726			TYPE 1131
APR 68	419675			IBM AA663

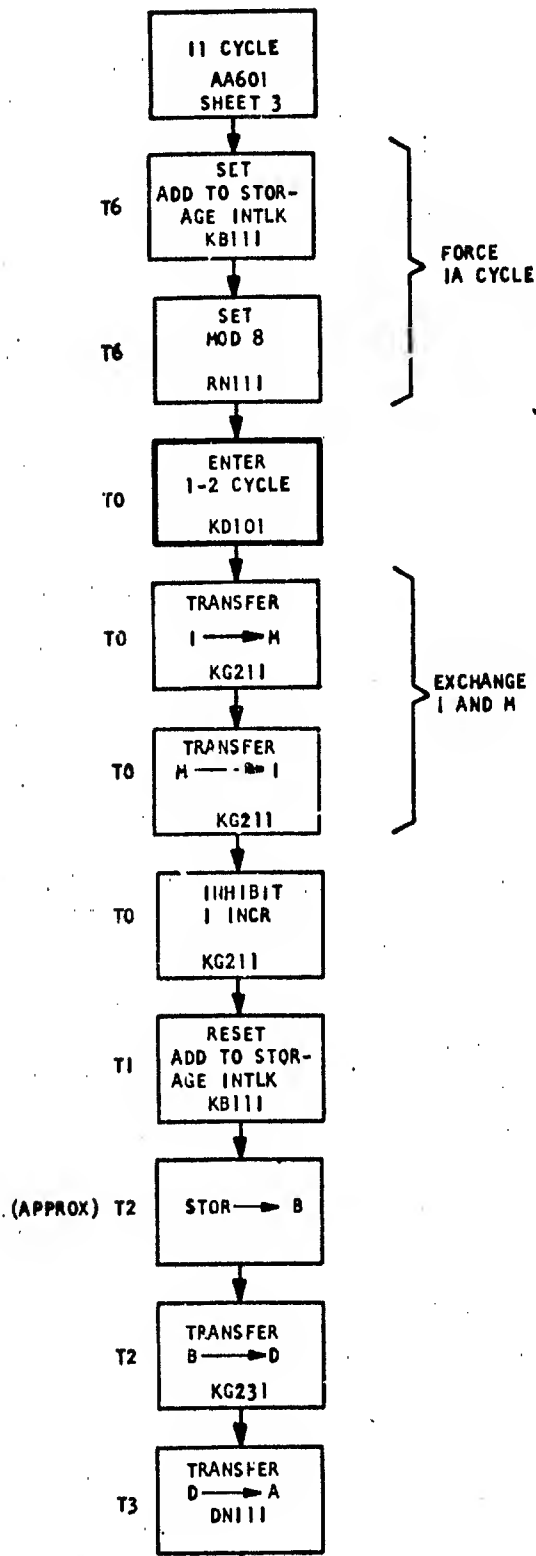
OP CODE 01110 MODIFY INDEX &
SKIP - FORMAT = 1 TAG = 00
(ADD TO STORAGE)

SHEET 3

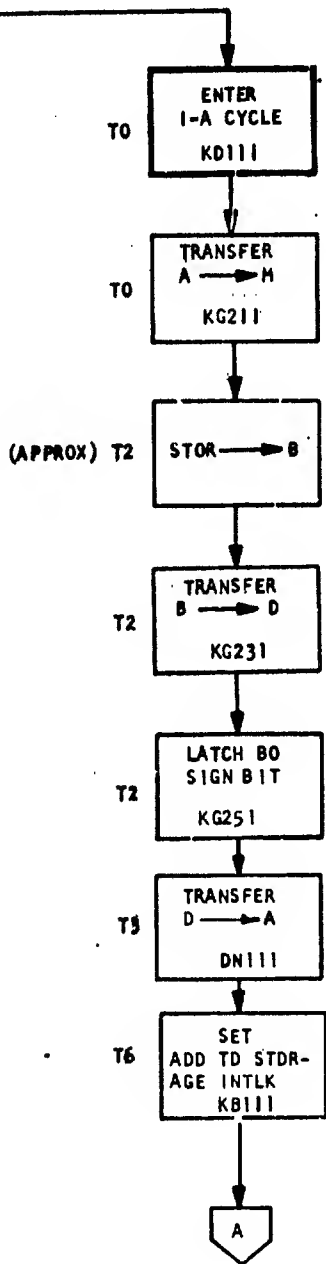
OBJECTIVE:

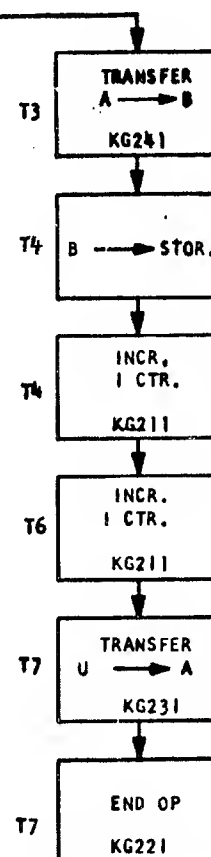
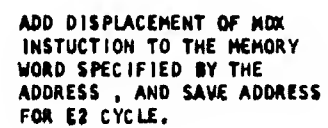
1. ADD THE DISPLACEMENT TO THE CONTENTS OF THE MEMORY LOCATION SPECIFIED BY THE ADDRESS PORTION OF THE INSTRUCTION.
2. SKIP NEXT ONE WORD INSTR. IF MODIFIED WORD CHANGES SIGN OR GOES THRU ZERO.

OP CODE 01110 MODIFY INDEX			
6 SKIP-FORMAT=1 TAG=00			
DATE	EC NUMBER	DATE	EC NUMBER
OCT 65	415480D	5-24-65	2201448
JAN 66	415483A	P/N	TYPE
APR 68	415726		1131
	419675		
IBM			
AA663			



TRANSFER THE ADDRESS PORTION OF THE INSTRUCTION
TO THE ACCUM AND SAVE MDX INSTRUCTION ADDRESS IN
IAR FOR SUBSEQUENT USE DURING E1 CYCLE.





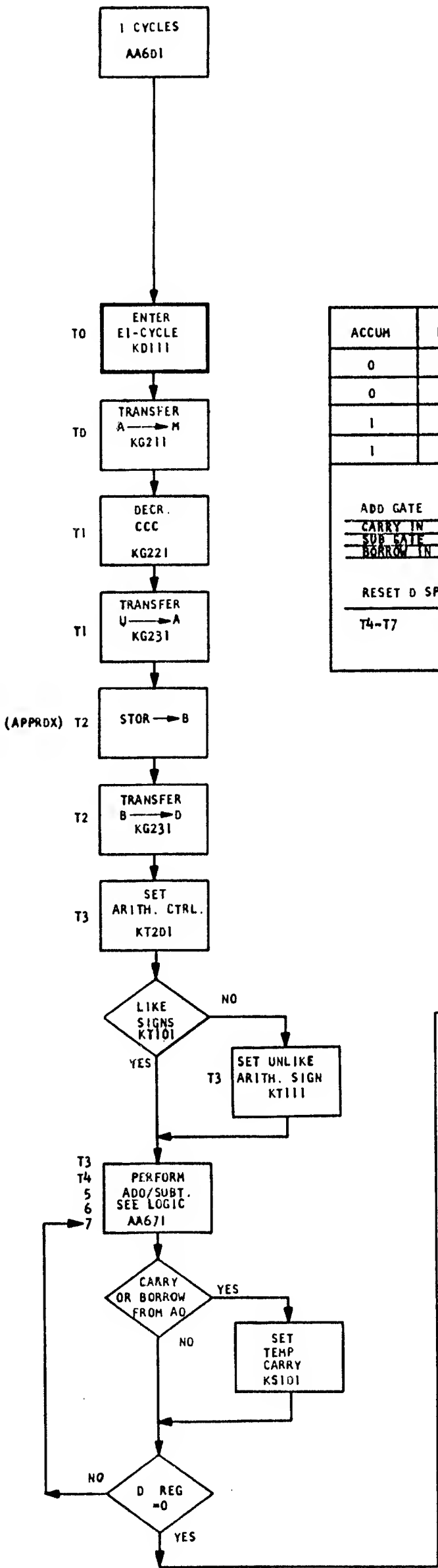
STORE MODIFIED WORD
AND INCREMENT I
COUNTER TO NEXT
INSTR.

DATE	EC NUMBER	DATE	EC NUMBER	OP CODE 01110	MODIFY INDEX
	415480D			6 SKIP FORMAT=1 TAG=00	
OCT 65	415483A			DATE	5-24-65 P/N 2201448
JAN 66	415726				TYPE 1131
APR 68	419675			IBM AA663	

7
6
5
4
3
2

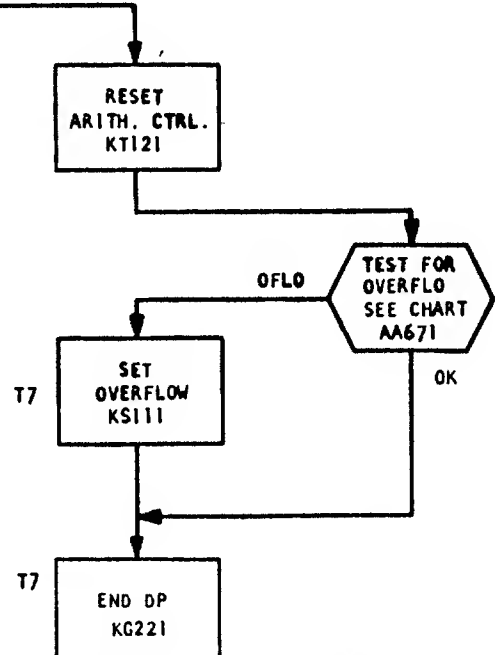
OP CODE 10000 ADD/
OP CODE 10010 SUBT

- OBJECTIVES:
- 1. ADD (OR SUBT.) THE CONTENTS OF THE EFF. ADDR. TO THE ACCUM. THE RESULT WILL BE IN THE ACCUM.
 - 2. TURN ON OVERFLO INDICATOR:
IF SUM > $2^{15} - 1$
IF DIFF < -2^{15}
 - 3. SET CARRY IF CARRY OR BORROW IS DETECTED OUT OF AO.



ADD/SUBTRACT LOGIC

ACCUM	D REG	SUM	CARRY	BORROW	ACTION ON RESET D REG S.P.
		DIFF			
0	0	0	0	0	NO ACTION
0	1	1	0	1	RESET D, SET A, GENERATE BORROW
1	0	1	0	0	NO ACTION
1	1	0	1	0	RESET D, RESET A, GENERATE CARRY



ADD/SUB OVERFLO

ACCUM RESULT NEG(AO=1)	TEMP CARRY OR BORROW	ARITH SIGN (UNLIKE)	OVERFLO
POS	0	LIKE	NO
POS	D	UNLIKE	YES
POS	I	LIKE	YES
POS	I	UNLIKE	NO
NEG	D	LIKE	YES
NEG	D	UNLIKE	NO
NEG	I	LIKE	NO
NEG	I	UNLIKE	YES

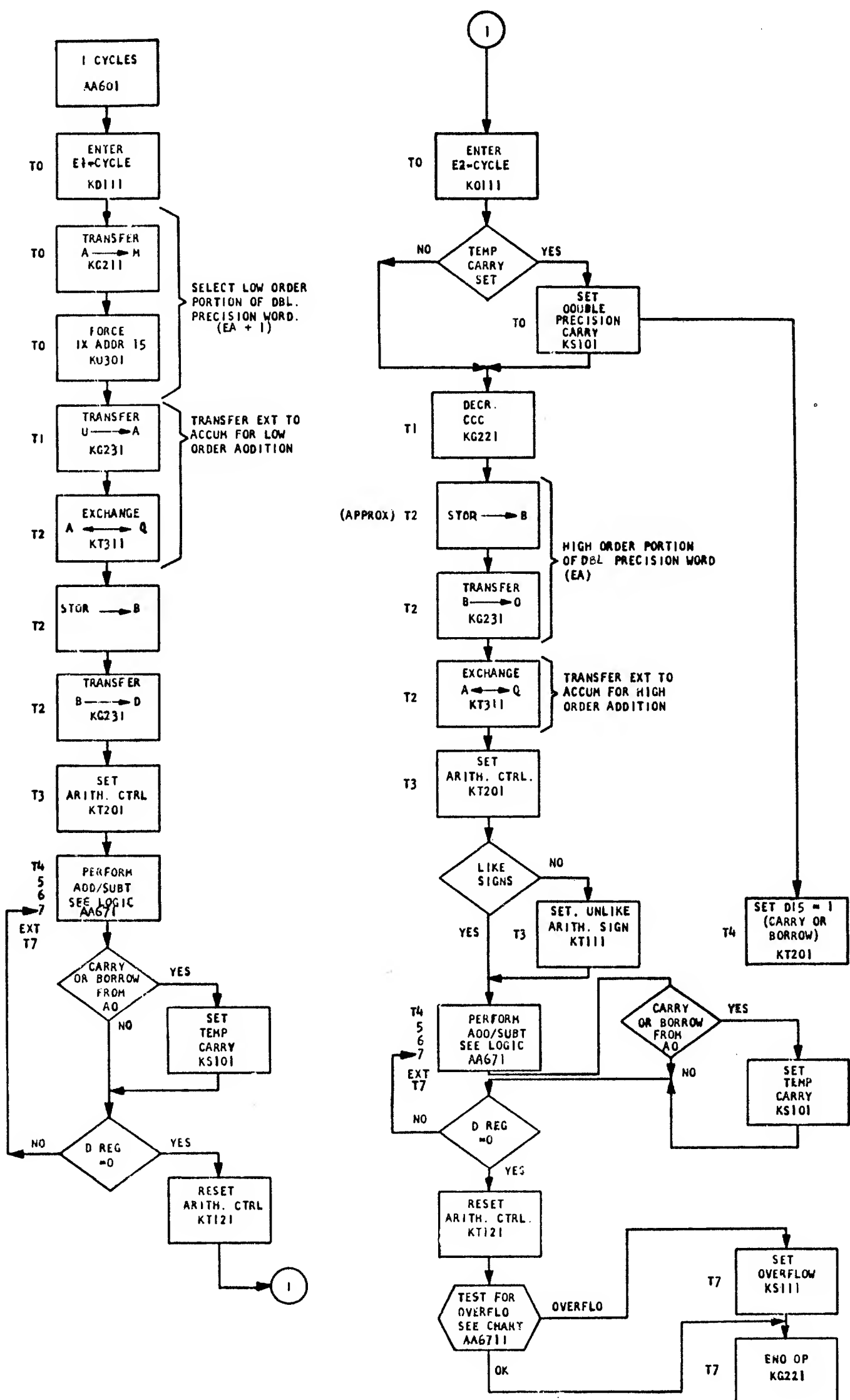
OP CODE 10000 ADD	EC NUMBER	DATE	EC NUMBER	DATE	P/N	TYPE	AA671
OP CODE 10010 SUBTRACT	4154800				2201450		
	419675					1131	
							IBM

OP CODE 10001 DBL PRECISION ADD
OP CODE 10011 DBL PRECISION SUBT

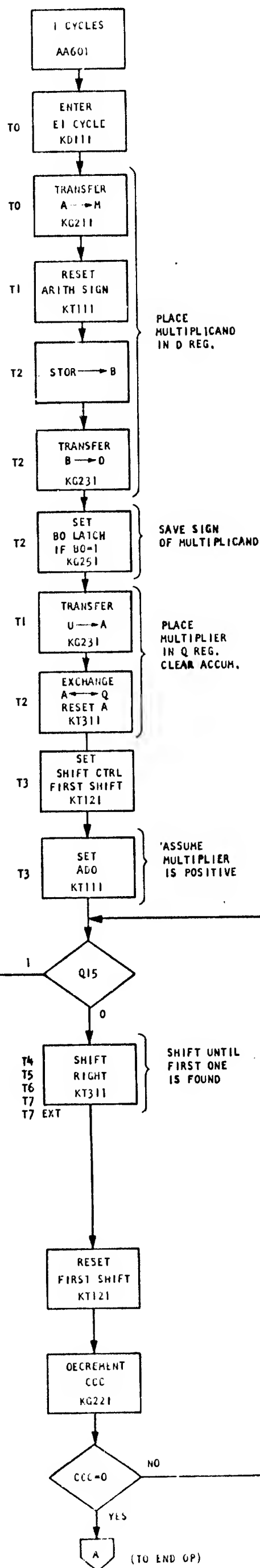
OBJECTIVES:

1. ADD OR SUBTRACT THE CONTENTS OF EA AND EA + 1 TO THE ACCUM AND THE ACCUM EXT (QREQ) AS ONE DOUBLE PRECISION (32 BIT) WORD.
2. EA MUST BE EVEN FOR CORRECT OPERATION.
3. SET CARRY AND OVERFLO AS IN SINGLE ADD OR SUB.

OP CDE 10001 DBL PRECISION ADD		OP CDE 10011 DBL PRECISION SUBT	
DATE	EC NUMBER	DATE	EC NUMBER
OCT 65	415480E	OCT 65	415480E
JAN 66	415483A	JAN 66	415483A
22APR68	419675	22APR68	419675
P/N 2201451		P/N 2201451	
TYPE 1131		TYPE 1131	
AA672		AA672	



OP CODE 10100		MULTIPLY		DATE		TYPE		AA673	
EC NUMBER	DATE	EC NUMBER	DATE	DATE	P M	DATE	TYPE	IBM	
				OCT 65	415483A				
				22APR68	419675				



OBJECTIVE:

- MULTIPLY CONTENTS OF EFFECTIVE ADDRESS BY THE CONTENTS OF THE ACCUMULATOR.
- THE RESULT WILL BE A 32 BIT DOUBLE PRECISION PRODUCT LOCATED IN THE ACCUM AND EXT.
- THERE IS NO CARRY OR OVERFLOW IN MULTIPLY.

ALGORITHM:

- RAPID MULTIPLICATION DEPENDS ON THE FACT THAT ANY BINARY NUMBER MAY BE REPRESENTED BY POWERS OF TWO AS FOLLOWS.

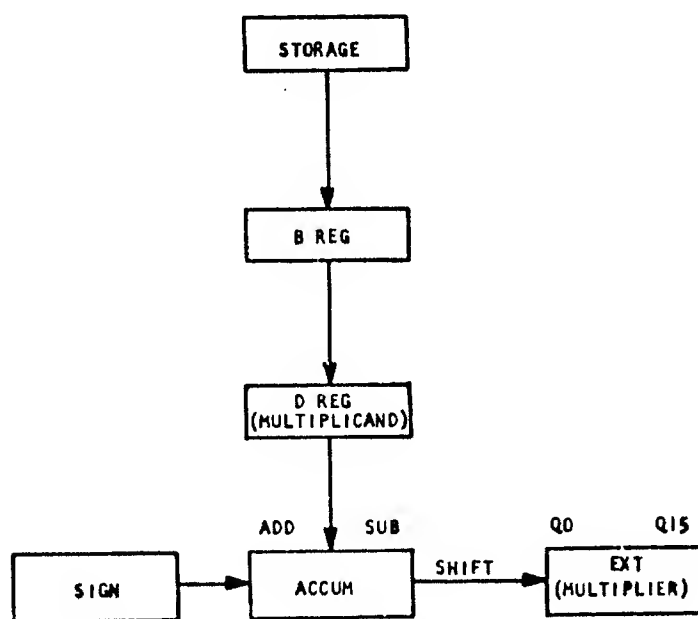
POW OF 2	8 7 6 5 4 3 2 1 0	RAPID EQUIV	LONG EQUIVALENT
BINARY	0 0 1 1 1 1 0 1 0	$2^7 - 2^3 + 2^1$	$2^6 + 2^5 + 2^4 + 2^3 + 2^1$
BINARY	0 0 1 1 1 0 1 1 1	$2^7 - 2^3 - 2^0$	$2^6 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0$

- THUS IT IS NOT NECESSARY TO FORM THE PARTIAL PRODUCT BY ADDING FOR EACH BIT POSITION. WE MAY EXAMINE THE MULTIPLIER (TWO LOWEST ORDER BITS AT A TIME) TO DETERMINE WHEN TO ADD THE MULTIPLICAND, WHEN TO SUBTRACT, OR WHEN TO JUST SHIFT THE MULTIPLIER.

MULTIPLIER Q14	Q15	PREVIOUS ACTION	NEW ACTION	EXPLANATION
0	0	ADD	SHIFT	NO ACTION
0	1	ADD	ADD, SHIFT	SINGLE ONE IN STRING OF ZEROS
1	0	ADD	SHIFT	NO ACTION
1	1	ADD	SUB, SHIFT	START STRING OF ONES
0	0	SUB	ADD, SHIFT	END OF STRING OF ONES
0	1	SUB	SHIFT	NO ACTION
1	0	SUB	SUB, SHIFT	SINGLE ZERO IN STRING OF ONES
1	1	SUB	SHIFT	NO ACTION

- THIS ALGORITHM PERMITS THE 1130 TO USE FEWER ADD CYCLES THAN WOULD BE POSSIBLE WITH CONVENTIONAL MULTIPLY.

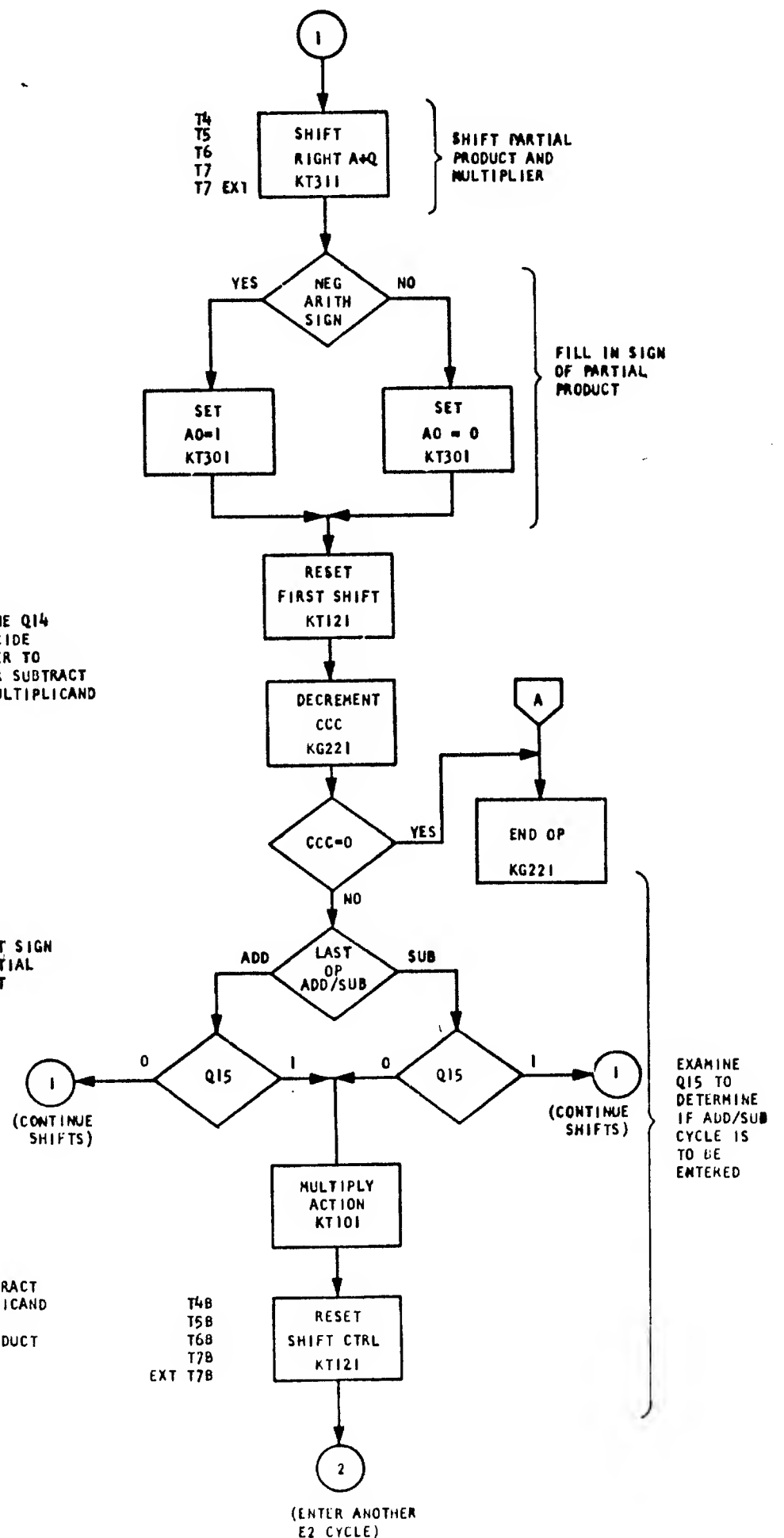
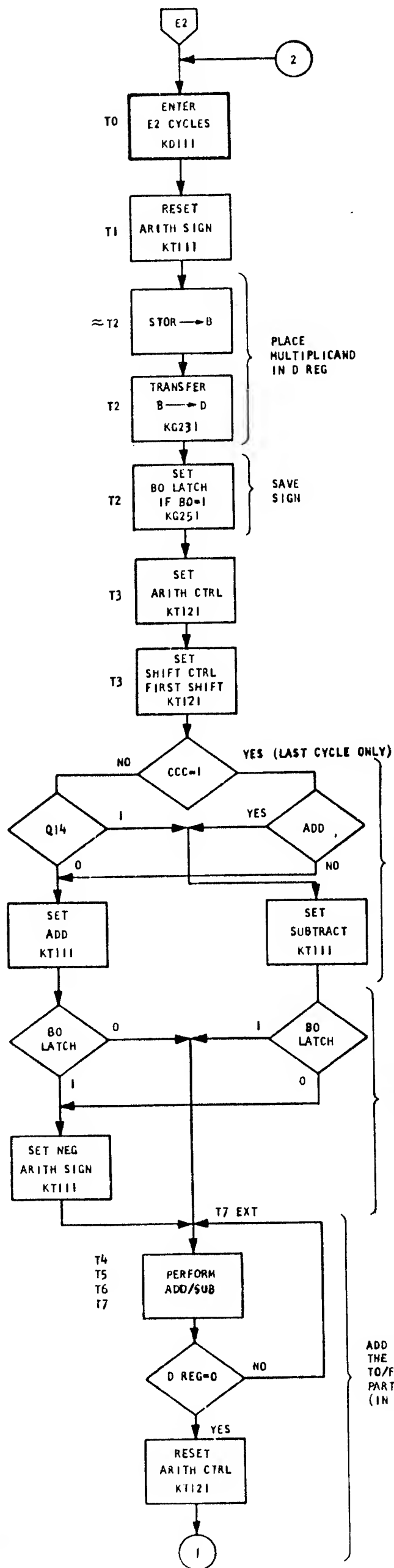
4. DATA FLOW.



ALGORITHM (CONT):

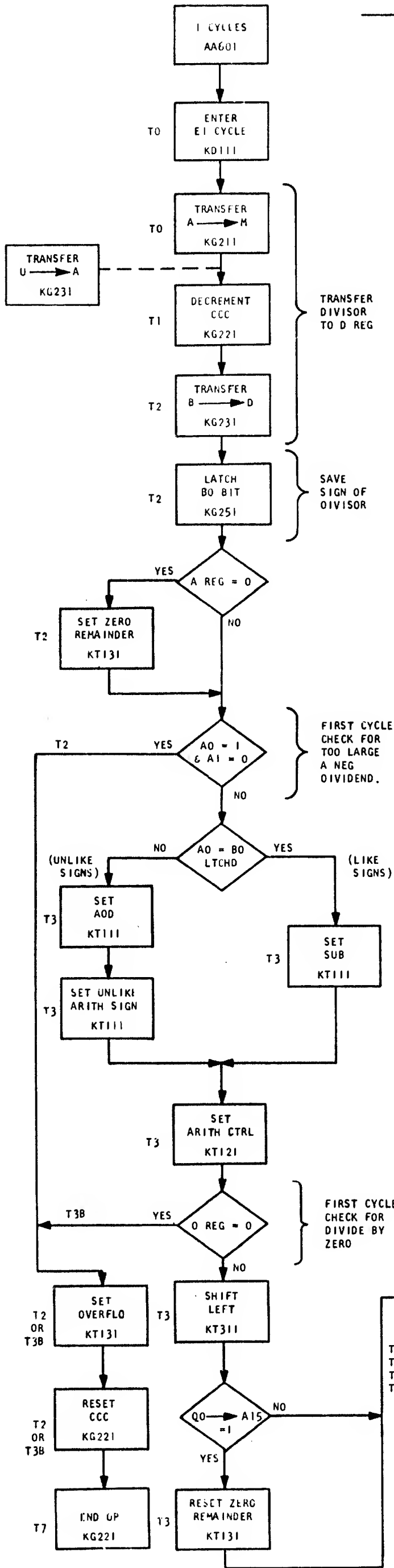
5. MULTIPLY E2 CYCLES ARE ENTERED WHEN AN EXAMINATION OF THE Q15 BIT INDICATES THAT IT IS DESIRED TO ADD OR SUBTRACT THE MULTIPLICAND TO/FROM THE ACCUMULATOR PARTIAL PRODUCT.
6. IN THE FIRST PART OF THE E2 CYCLE THE Q14 BIT IS EXAMINED TO DETERMINE WHETHER ADDITION OR SUBTRACTION IS DESIRED.
7. IN THE SECOND PART OF THE E2 CYCLE SHIFTING IS CONTINUED UNTIL THE Q15 BIT INDICATES THAT ARITHMETIC ACTION IS AGAIN REQUIRED, OR UNTIL THE CCC COUNT INDICATES THAT ALL SIXTEEN BITS HAVE BEEN EXAMINED (CCC=0).

OP CODE 10100		MULTIPLY		DATE		P/N		TYPE		AA673	
DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER
OCT 65	415483A										
22APR68	419675										



OP CODE 10101 DIVIDE
FIRST CYCLE

SHEET 1



OBJECTIVES:

- 1) THE CONTENTS OF THE ACCUMULATOR AND THE Q REG (TREATED AS A 32 BIT DOUBLE PRECISION WORD) ARE DIVIDED BY THE CONTENTS OF THE EFFECTIVE ADDRESS.
- 2) AT THE END OF THE OPERATION, THE QUOTIENT WILL BE FOUND IN THE ACCUMULATOR AND THE REMAINDER IN THE Q REGISTER.

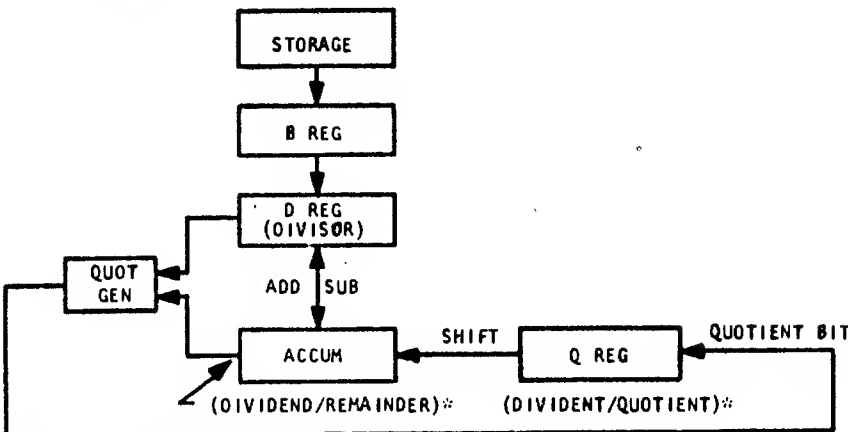
ALGORITHM:

- 1) THE QUOTIENT OF TWO BINARY NUMBERS MAY BE GENERATED BY SUCCESSIVELY SHIFTING AND SUBTRACTING THE DIVISOR FROM APPROPRIATE ORDERS OF THE DIVIDEND. IF THE SUBTRACTION WAS SUCCESSFUL (REMAINDER SIGN BIT SAME AS DIVISOR SIGN BIT) A QUOTIENT ONE BIT IS GENERATED AND ANOTHER REDUCTION CYCLE IS TRIED. IF THE SUBTRACTION WAS NOT SUCCESSFUL (REMAINDER SIGN BIT NOT THE SAME AS DIVISOR SIGN BIT) A QUOTIENT ZERO BIT IS GENERATED AND AN ADDITION CYCLE IS TAKEN TO RESTORE THE REMAINDER.
- 2) THIS SHIFT-ADD/SUB PROCEDURE MAY BE ILLUSTRATED AS FOLLOWS: DIVIDE BINARY 00111001 (57) BY 0101 (5)

SHIFT & SUB:	0 0 1 1 1 0 0 1	
	0 1 0 1	QUOTIENT
	+ 0 0 0 1 0 0 0 1	1
SHIFT & SUB:	0 1 0 1	
	- 1 1 1 1 1 1 0 1	0
SHIFT & ADD:	0 1 0 1	
	0 0 0 0 1 1 1 1	1
SHIFT & SUB:	0 1 0 1	
	+ 0 0 0 0 0 0 1 0	1

ANSWER: QUOTIENT 1011 (11) REMAINDER 0010 (2).

3) DATA FLOW



*NOTE THAT A AND Q ARE EXCHANGED AT END OF OP.

OP CODE 10101 DIVIDE		P/N		AA674	
DATE	EC NUMBER	DATE	TYPE	IBM	
OCT 65	415483A	AUG 65	1130		
22APR68	419675				



- 4) THE DIVIDE INSTRUCTION CONSUMES EIGHTEEN CYCLES (ONE E1 AND SEVENTEEN E2). DURING THE FIRST SIXTEEN CYCLES THE QUOTIENT IS BUILT UP BY THE SHIFT-ADD/SUB PROCESS. THE SEVENTEENTH AND EIGHTEENTH CYCLE ARE USED FOR CORRECTION AND CHECKING AS EXPLAINED ON SHEET 3.

5) DVERFLDW:

THE PURPOSE OF DIVIDE OVERFLOW IS TO DETECT
QUOTIENT OVERFLOW CONDITIONS RESULTING FROM
A DIVIDEND WHICH IS TOO LARGE IN RELATION TO
THE DIVISOR. THIS MAY BE BROKEN DOWN AS
FOLLOWS:

5.1 FIRST CYCLE CHECKS -

```

CHECK FDR ZERD DIVISDR
CHECK FDR TOO LARGE A NEGATIVE DIVIOEND

```

5.2 CHECKS AFTER FIRST REDUCTION CYCLE -

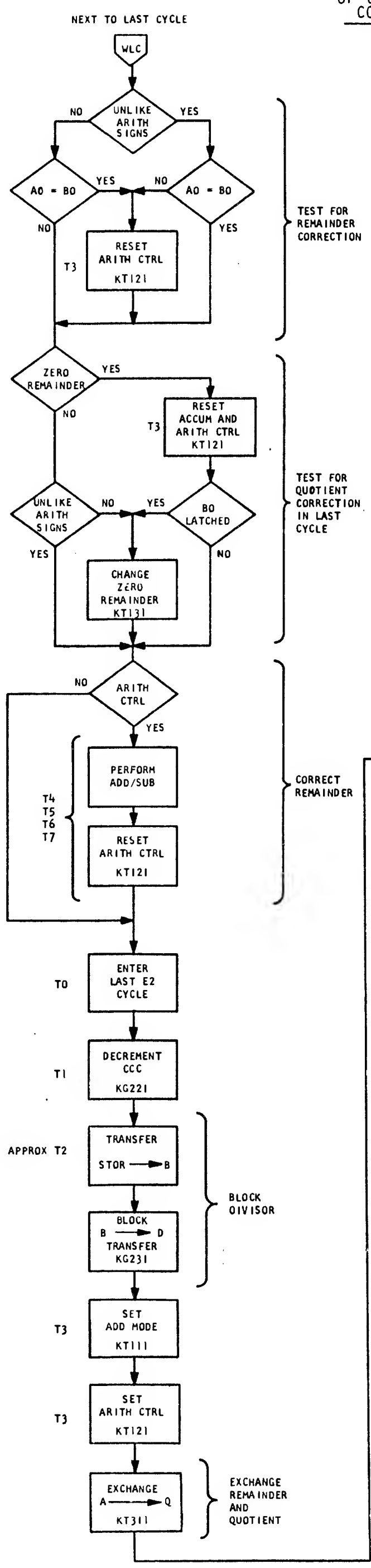
CHECKS FOR A REMAINDER WHICH IS TOO LARGE TO BE REPRESENTED CORRECTLY IN THE ACCUMULATOR (SIMILAR TO ADD/SUB OVERFLOW). CHECKS FOR EXCEPTIONAL CASES (OF UNLIKE DIVIDEND AND DIVISOR SIGNS), WHICH ARE NOT DETECTABLE BY THE LAST CYCLE CHECK.

5.3 LAST CYCLE CHECK -

CHECKS THAT LIKE DIVIDEND AND DIVISOR
SIGNS GIVE A POSITIVE QUOTIENT. CHECKS
THAT UNLIKE DIVIDEND AND DIVISOR SIGNS
GIVE A NEGATIVE QUOTIENT.

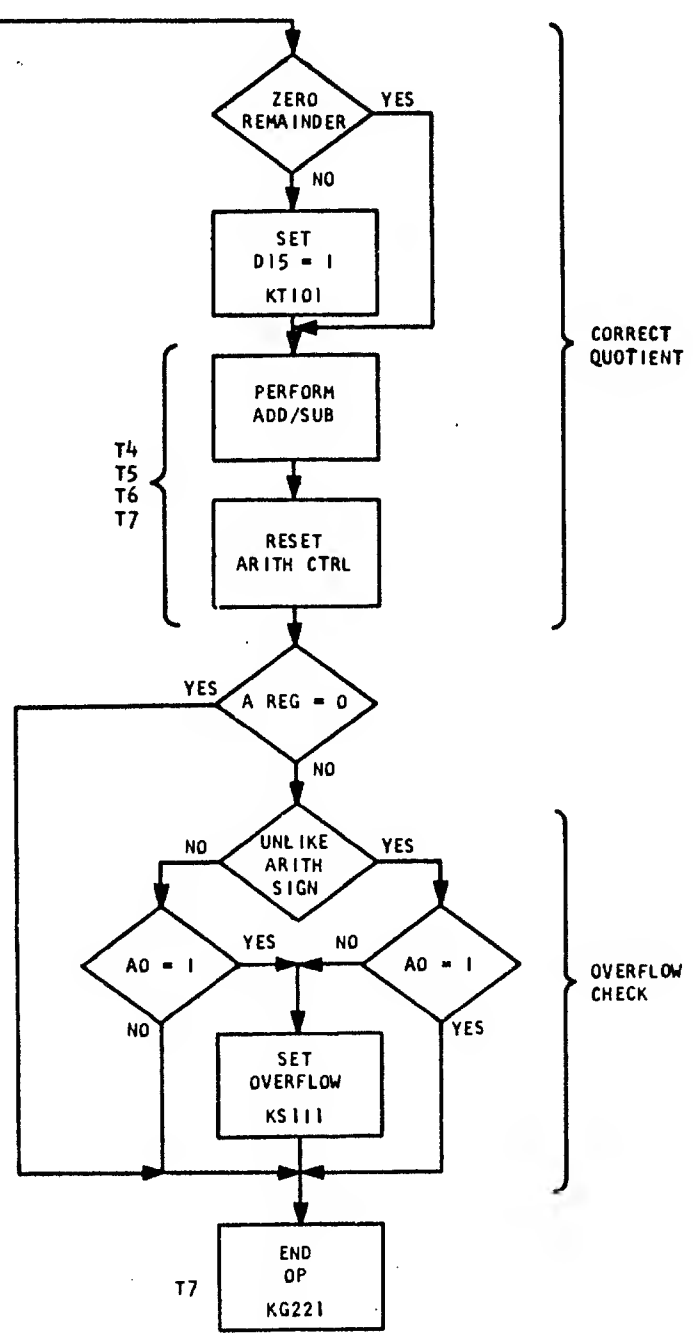
[illegible]

OP CODE 10101 DIVIDE		P/N		AA674	
DATE	EC NUMBER	DATE	P/N	TYPE	
OCT 65	415483A	AUG 65	2201453	1130	
22APR68	419675				

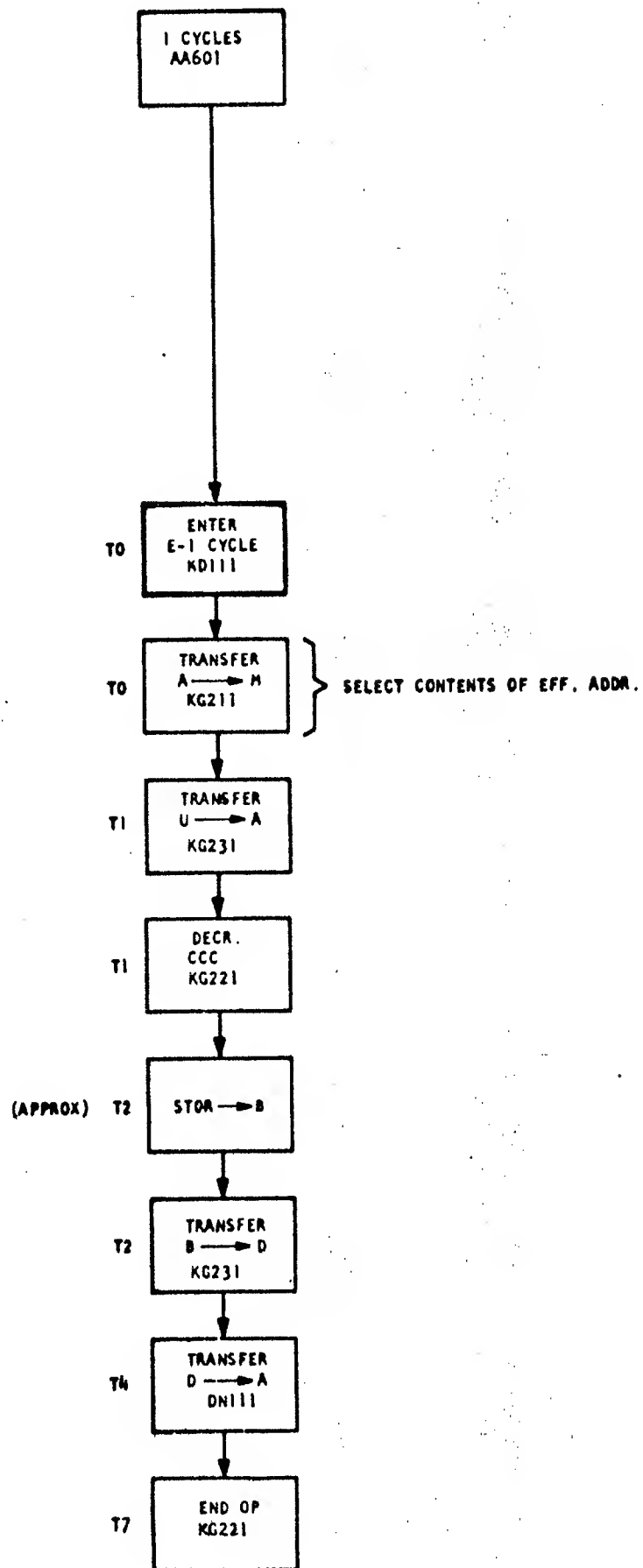


ALGORITHM (CONT.):

- 6) THE SEVENTEENTH (NEXT TO LAST) E CYCLE IS THE REMAINDER CORRECTION CYCLE. IF THE REMAINDER SIGN IS NOT THE SAME AS THE ORIGINAL DIVIOEND SIGN, THE DIVISOR IS ADDED TO (OR SUBTRACTED FROM) THE ACCUMULATOR TO CORRECT THE REMAINDER.
- 7) THE EIGHTEENTH (LAST) E CYCLE PLACES THE QUOTIENT IN THE ACCUMULATOR AND THE REMAINDER IN THE Q REGISTER, CORRECTS THE QUOTIENT, AND PERFORMS THE OVERFLOW CHECKS AS DESCRIBED UNDER 5.3. THE QUOTIENT GENERATED WILL BE EITHER A POSITIVE NUMBER, OR A NEGATIVE NUMBER IN ONE'S COMPLEMENT FORM. SINCE THE 1130 SYSTEM USES TWO'S COMPLEMENT REPRESENTATION FOR NEGATIVE NUMBERS, A ONE IS ADDED TO THE QUOTIENT WHEN A ONE'S COMPLEMENT RESULT IS DETECTED BY THE FOLLOWING:
- 7.1) DIVIOEND PLUS AND DIVISOR MINUS
- 7.2) DIVIOEND MINUS AND DIVISOR PLUS, EXCEPT WHEN REMAINDER IS ZERO
- 7.3) DIVIOEND MINUS AND DIVISOR MINUS, AND REMAINDER IS ZERO



TRANSFER THE CONTENTS OF THE EFF. ADDR.
INTO THE ACCUM.

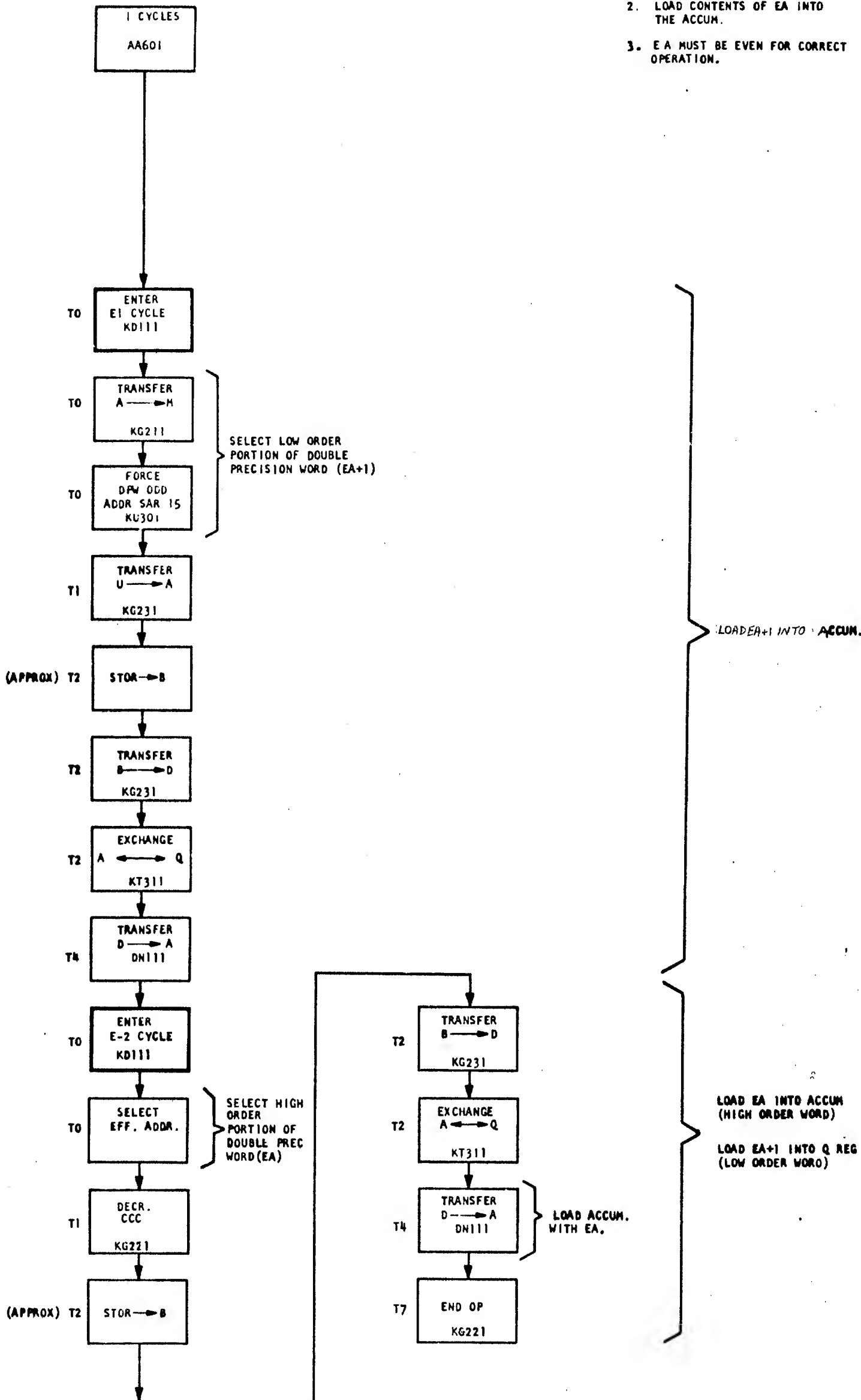


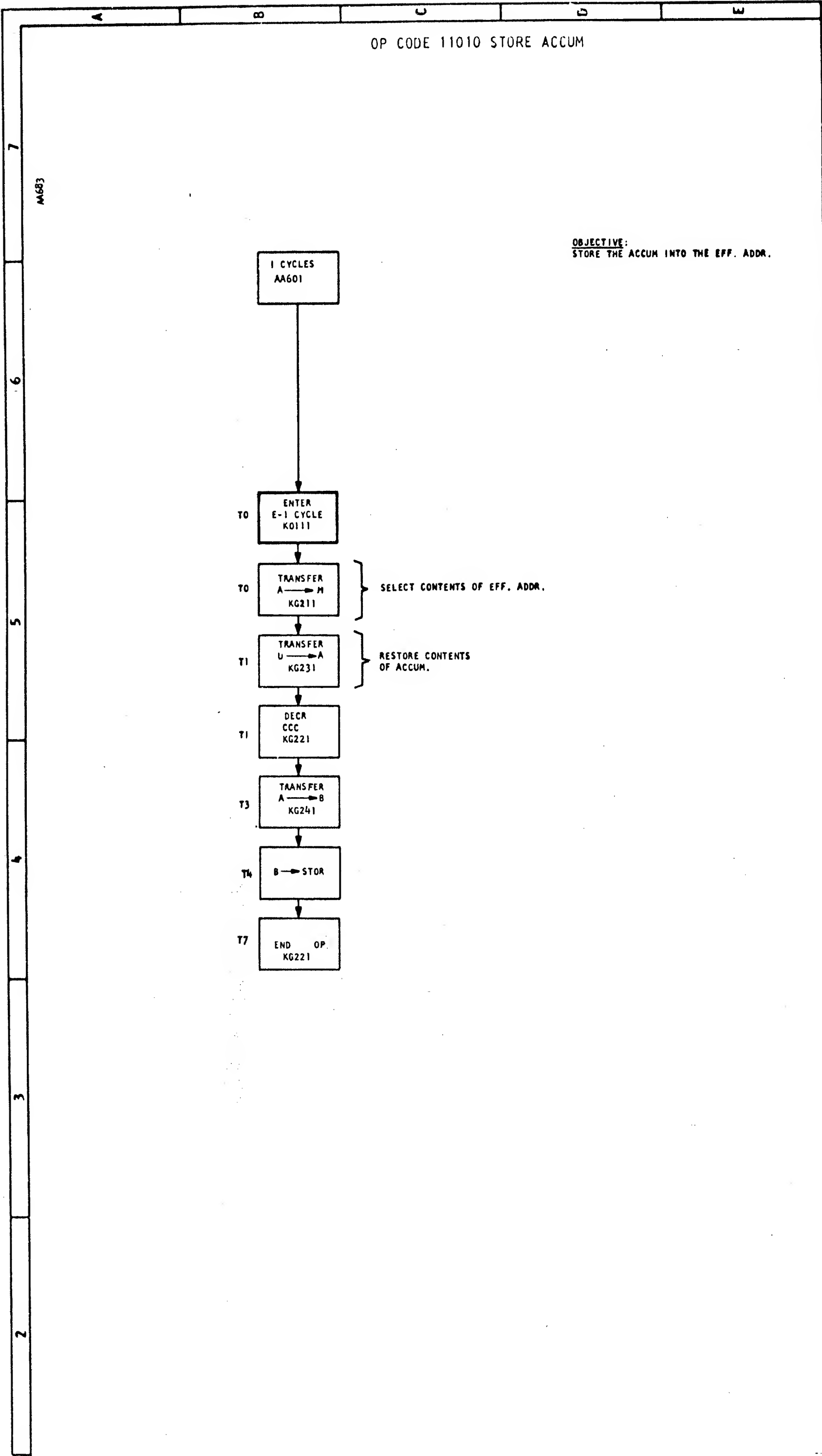
DATE	BC NUMBER	DATE	BC NUMBER	OP CODE	11000
	4154800			LOAD ACQU	
				DATE	5-24-65
					P M
					2201455
					TYPE
					1131
				IBM	AA681

OP CODE 11001 DOUBLE PREC LOAD

- OBJECTIVES:
- 1. LOAD CONTENTS OF EA+1 INTO THE Q REG.
 - 2. LOAD CONTENTS OF EA INTO THE ACCUM.
 - 3. EA MUST BE EVEN FOR CORRECT OPERATION.

OP CODE 11001		DOUBLE PREC LOAD		P/N		DATE		TYPE		AA682	
DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER	DATE	EC NUMBER
	4154800										
OCT 65	415483A										



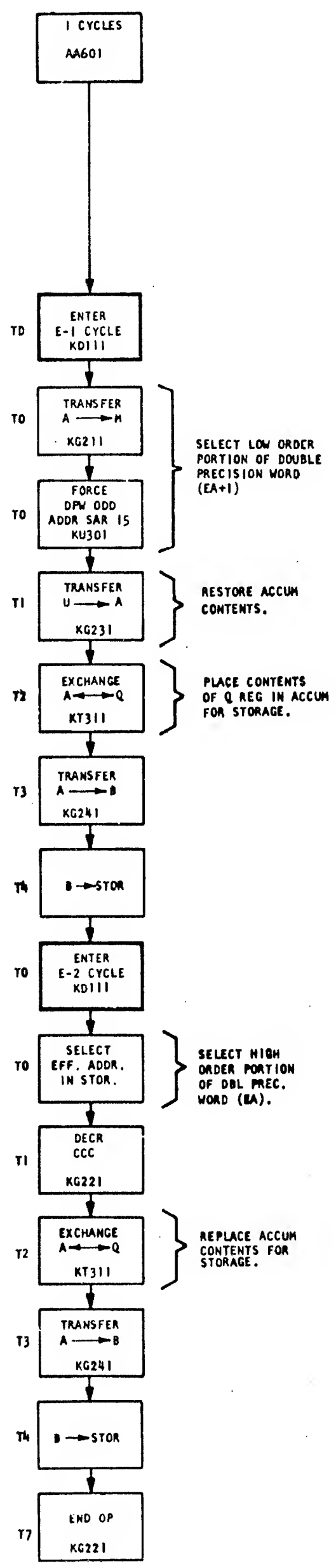


DATE		EC NUMBER	DATE	EC NUMBER	OP CODE 11010	
		4154800			STORE ACCUMULATOR	
					DATE	5-24-65
					P/N	2201457
					TYPE	1131
					PAGE NO	AA683
					IBM	

7
6
5
4
3
2

OP CODE 11011
DOUBLE PREC STORE

- OBJECTIVES:
- 1. STORE THE Q REG. INTO THE EFFECTIVE ADDRESS PLUS ONE. (E-1 CYCLE)
 - 2. STORE THE ACCUM INTO THE EFFECTIVE ADDRESS. (E-2 CYCLE)
 - 3. EA MUST BE EVEN FOR CORRECT OPERATION.



DATE		EC NUMBER		DATE		EC NUMBER	
OCT 65		415480D					
		415483A					

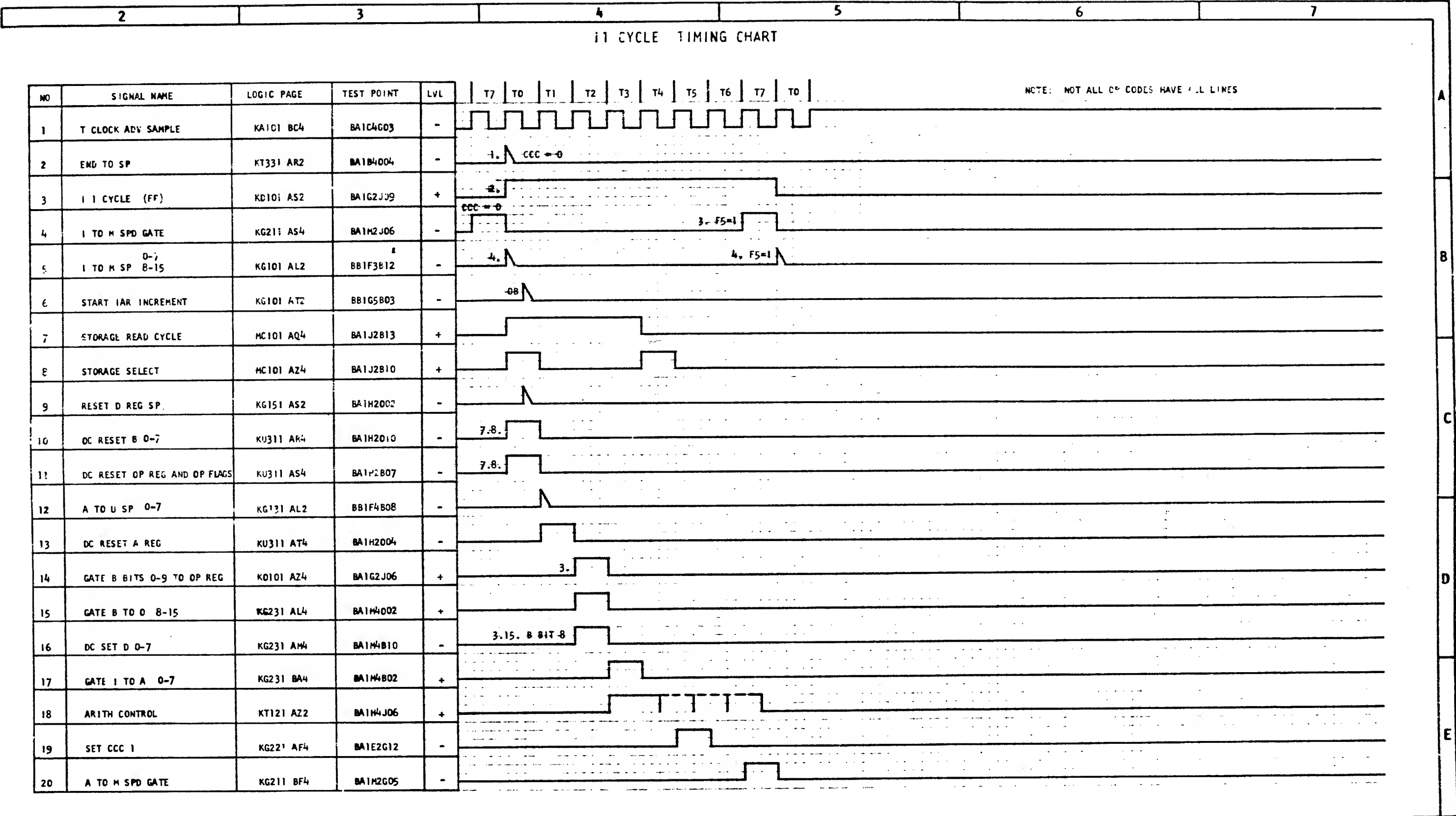


LOGICAL "OR"
OBJECTIVE:
2. THE CONTENTS OF THE EFF. ADDR. ARE "ORED"
BIT BY BIT WITH THE CONTENTS OF THE
ACCUM.

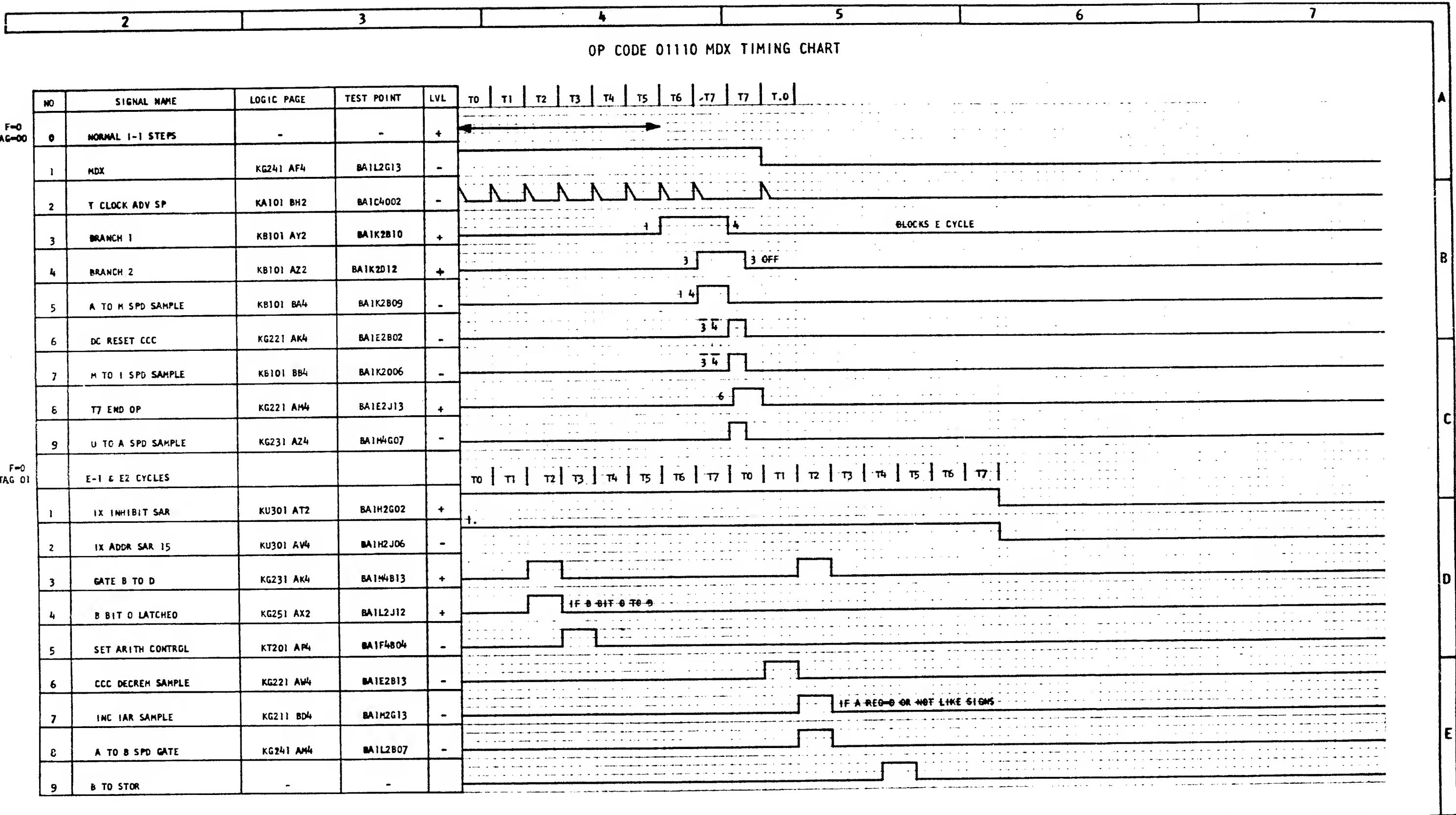
LOGICAL EXCLUSIVE "OR"
OBJECTIVES:
3. THE CONTENTS OF THE EFF. ADDR. ARE
EXCLUSIVE "ORED" BIT BY BIT WITH THE
CONTENTS OF THE ACCUM.

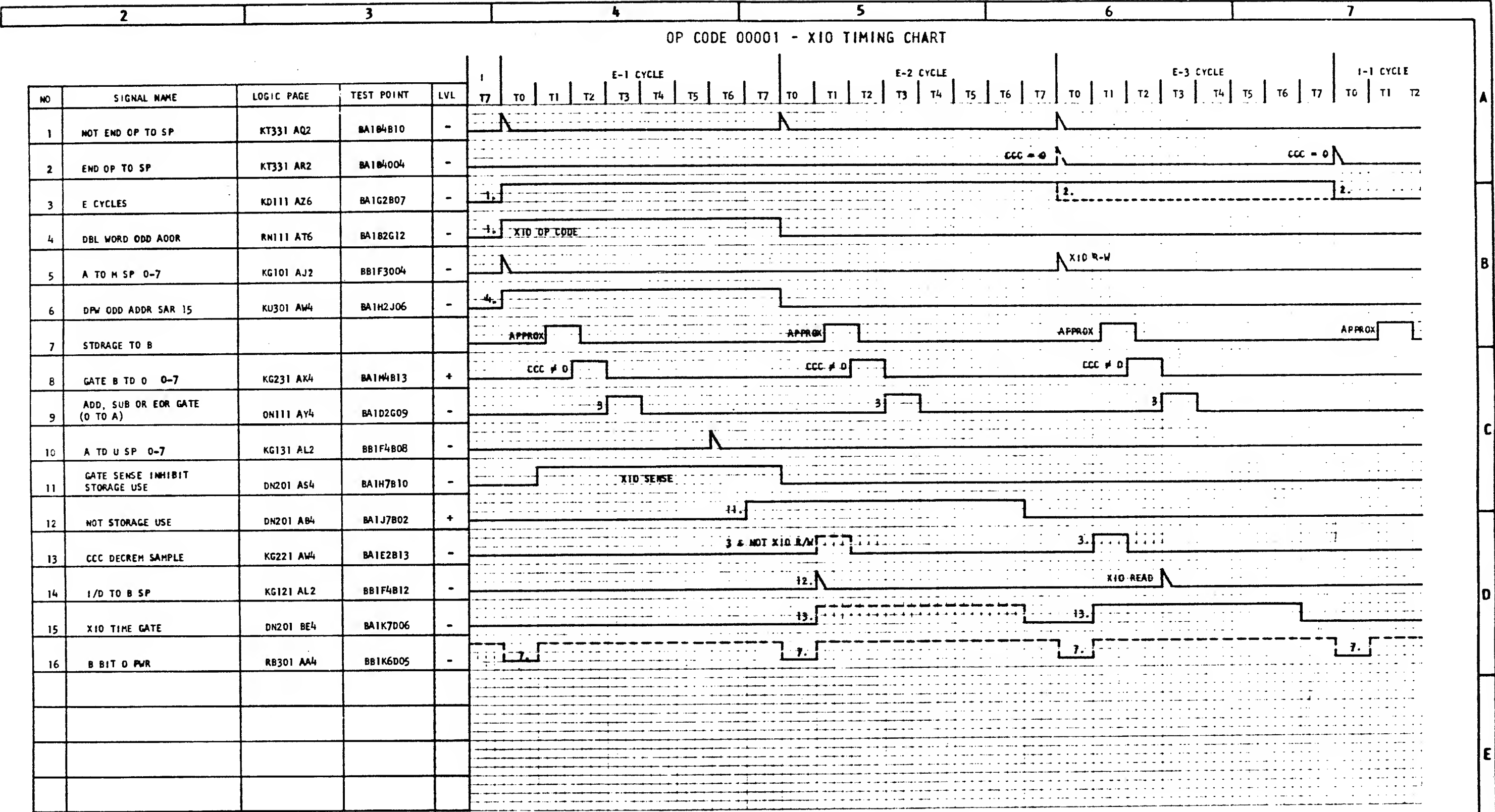
4. NOTE THAT NORMAL D \rightarrow A TRANSFERS ARE ACCOMPLISHED BY BRINGING UP BOTH 'AND' AND 'OR' GATES SIMULTANEOUSLY.

DATE	EC NUMBER	DATE	EC NUMBER	OP CDE 11100 LOGICAL AND
MAY -65	415480D			OP CDE 11101 LOGICAL OR OP CD
AUG -65	415480E			11110 LOGICAL EXCLUSIVE OR
				DATE 5-24-65 P/N 2201460
OCT 55	415483A			TYPE 1131
				IBM AA691

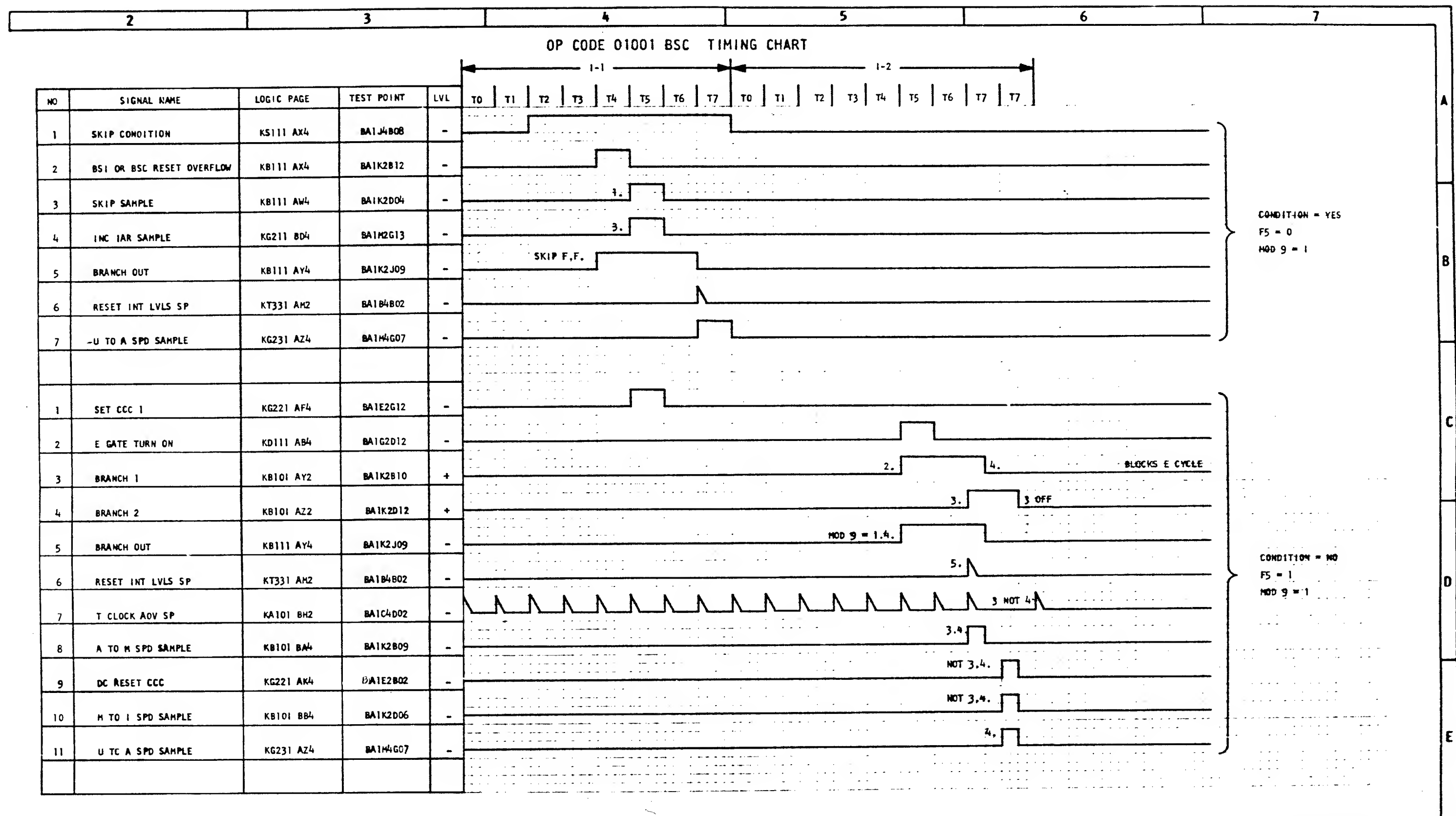


DATE	EC NUMBER	DATE	EC NUMBER	11 CYCLE TIMING CHART			
0C1 65	415483A						
22APR68	419675			DATE		P N	2201299
						TYPE	1130
				IBM		AA701	

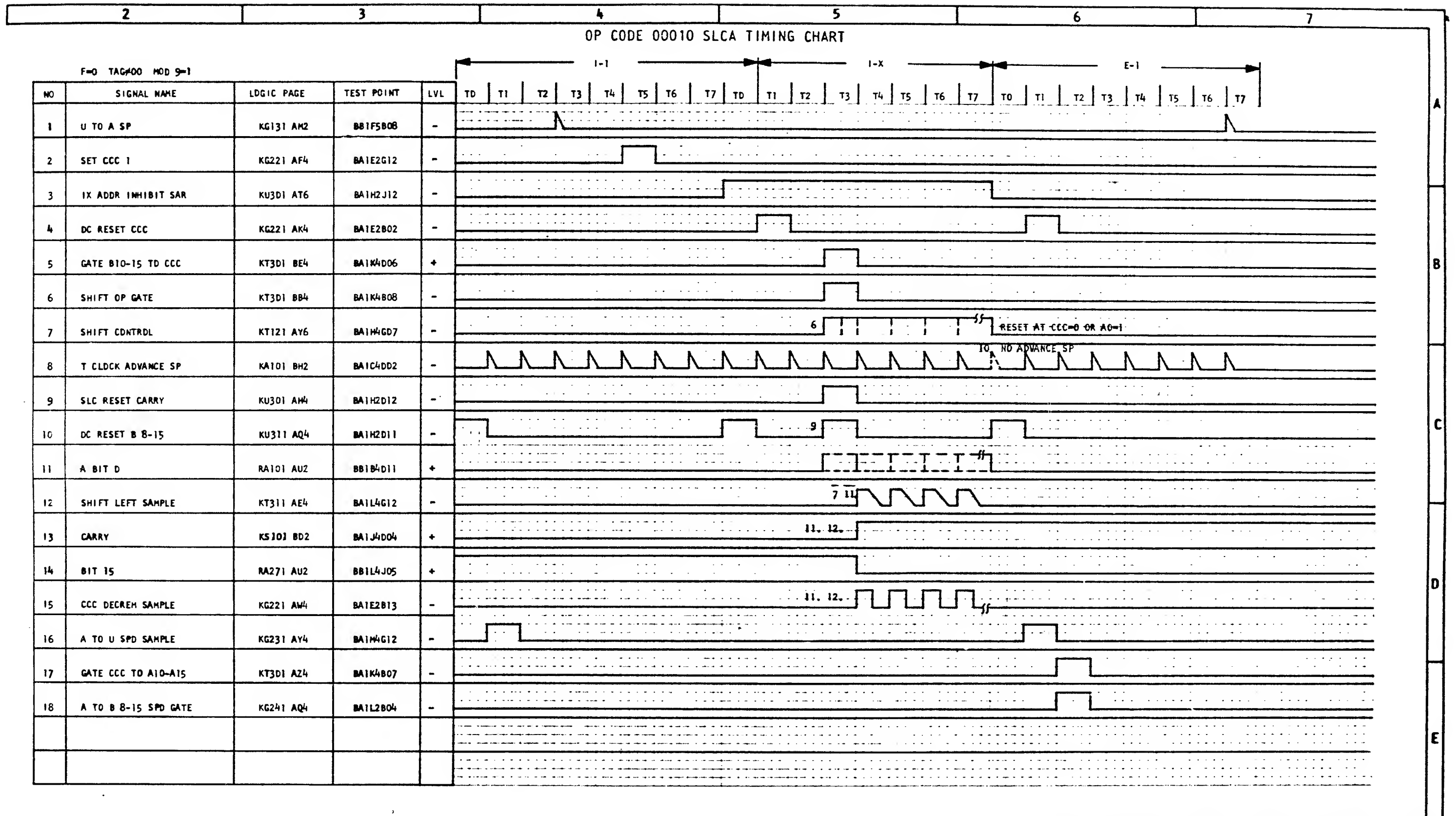




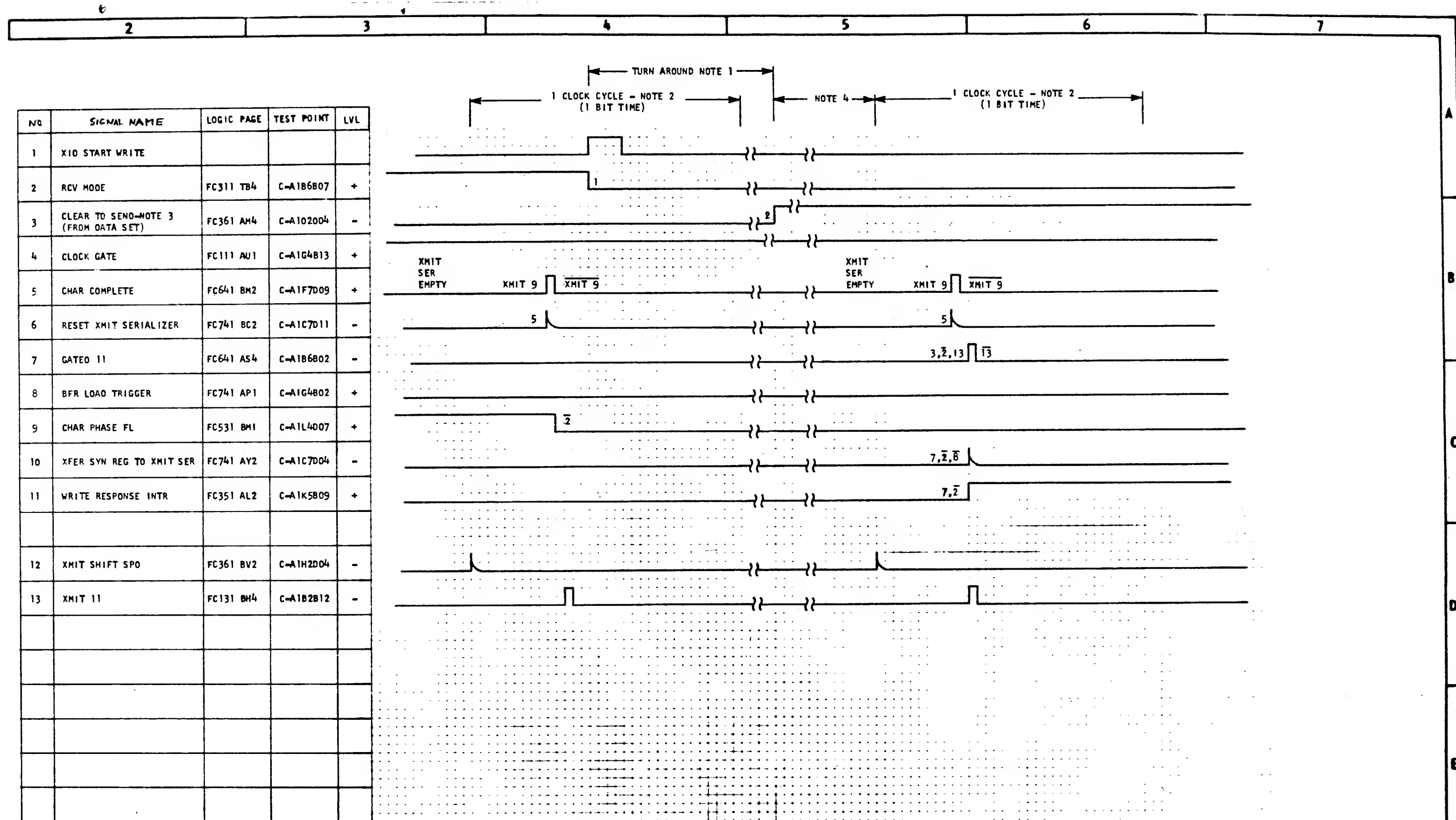
DATE	EC NUMBER	DATE	EC NUMBER	OP CODE 00001 - X10			
OCT 65	415483A			TIMING CHART			
22APR68	419675			DATE		P/N	2201297
						TYPE	1130
				IBM		AA721	



DATE	EC NUMBER	DATE	EC NUMBER	OP CODE 01001 BSC			
OCT 65	415483A			TIMING CHART			
22APR68	419675			DATE		P N	2201340
						TYPE	1130
				IBM		AA731	



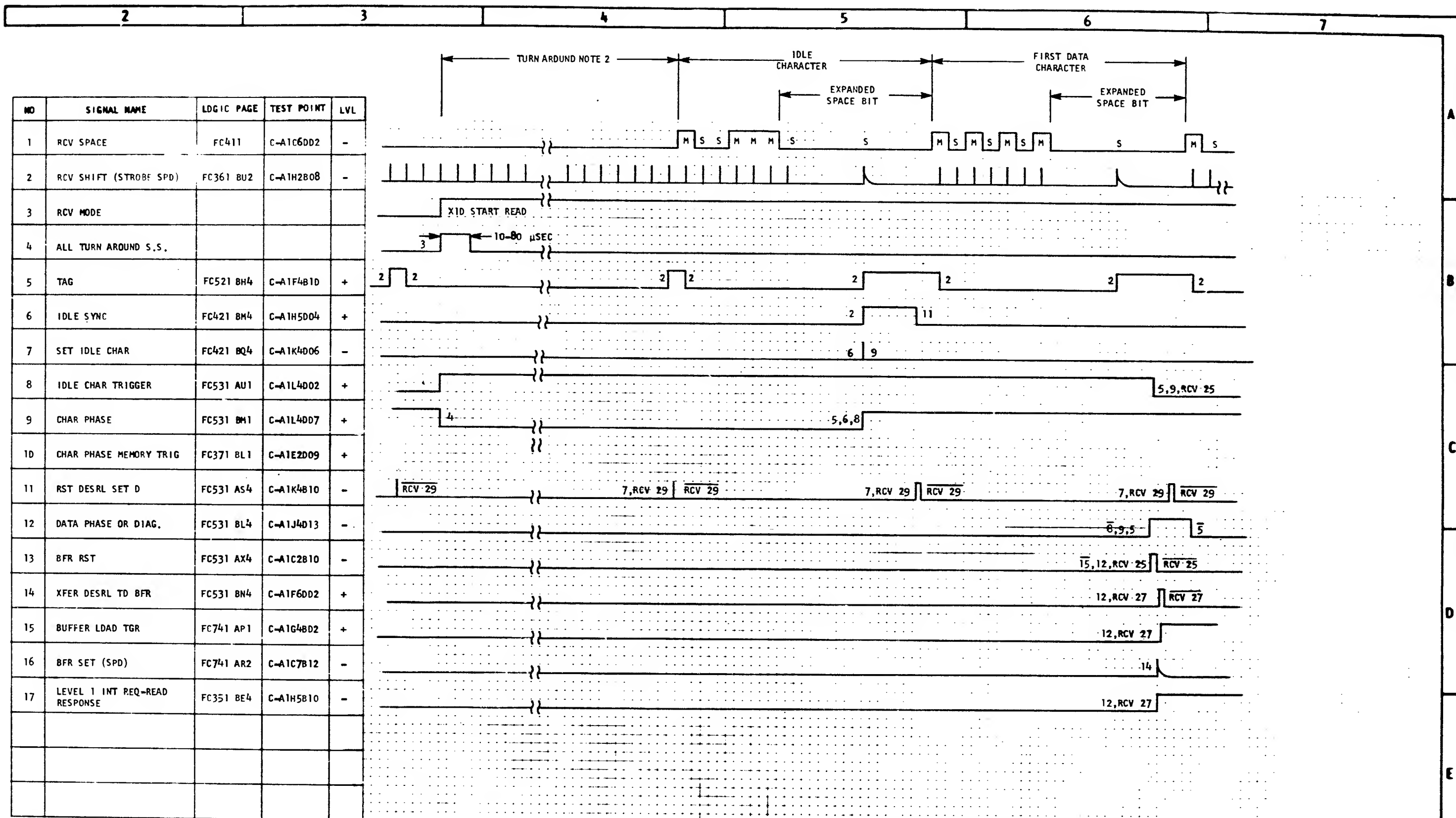
DATE	EC NUMBER	DATE	EC NUMBER	DP CODE 00010 SLCA			
OCT 65	415483A			TIMING CHART			
22APR68	419675			DATE		P-N	2201339
						TYPE	1130
				IBM		AA751	



NOTES:

- 200 MS TURNAROUND TIME FROM DATA SET IF 2 WIRE
- 1 CLOCK CYCLE TIME 600 BAUD IS 13.3 μ SEC
1200 BAUD IS 6.6 μ SEC
2000 BAUD IS 3.3 μ SEC
2400 BAUD IS 2.8 μ SEC
- CLEAR TO SEND IS ALWAYS ACTIVE DURING SCA OPERATION IF 4 WIRE
- TIME WILL VARY FROM ZERO TO MAXIMUM OF 8 BIT TIME (OR 6 OR 7 IF 6 OR 7 BIT FRAME) BECAUSE CLEAR TO SEND IS ASYNCHRONOUS TO XMIT CLOCK

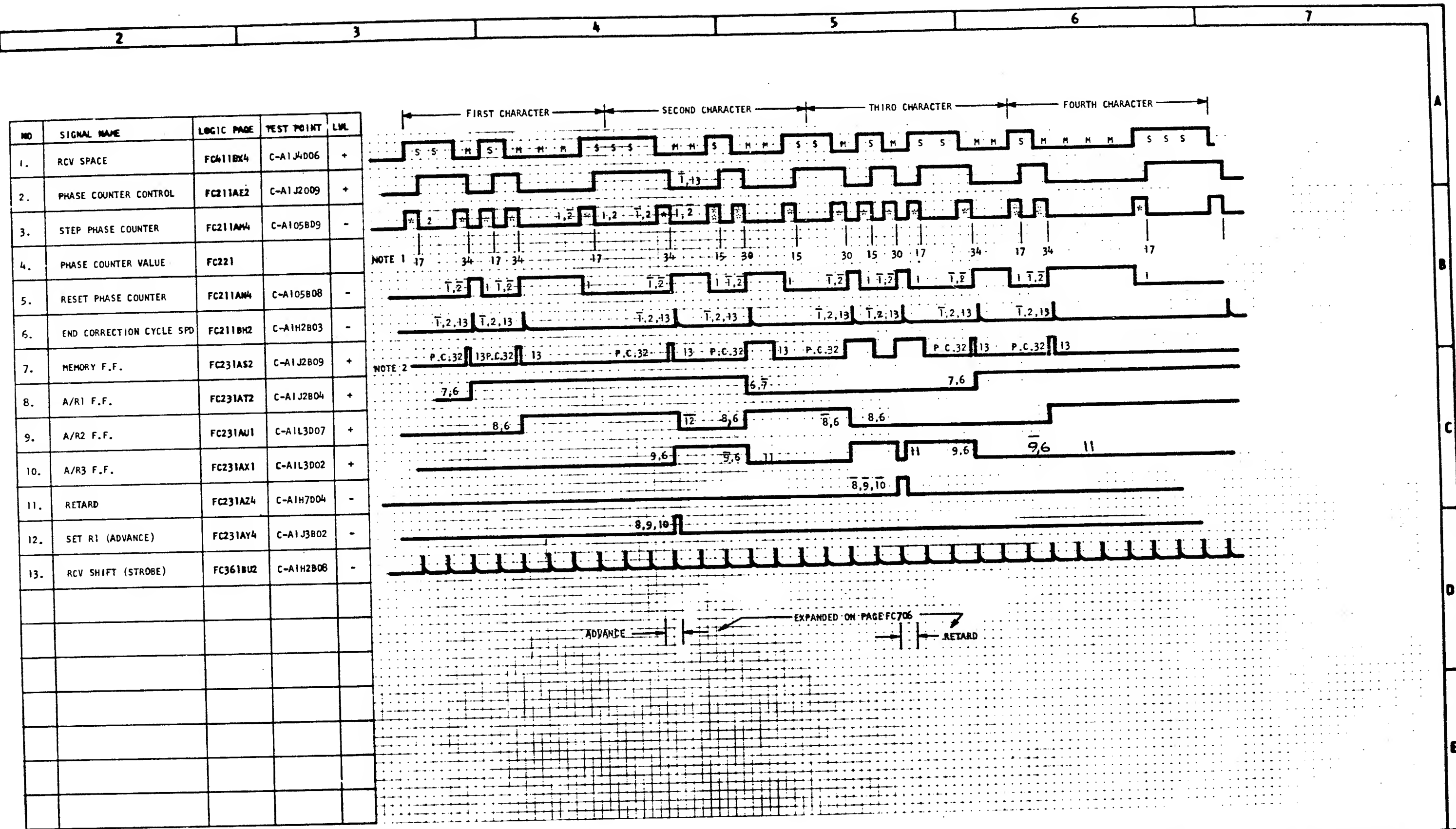
DATE	EC NUMBER	DATE	EC NUMBER	SCA START WRITE TIMING BSC.			
SEP 66	419632			STR. 2 OR 4 WIRE			
OEC 66	419644			DATE	P/W	2231301	
22APR68	419675				TYPE	1130	
				IBM		FC702	



NOTES:

1. BOTH STATIONS HAVE PREVIOUSLY ESTABLISHED SYNCHRONIZATION WITH THE EXCHANGE OF IDLE CHARACTERS. RECEIVING STATION IS SHOWN GOING FROM TRANSMIT MODE TO RECEIVE MODE TO RECEIVE THE BEGINNING OF A RECORD.
2. 200MS TURNAROUND TIME FROM DATA SET IF 2 WIRE TELEPHONE LINES.
ZERO MS TURNAROUND TIME FROM DATA SET IF 4 WIRE TELEPHONE LINES.

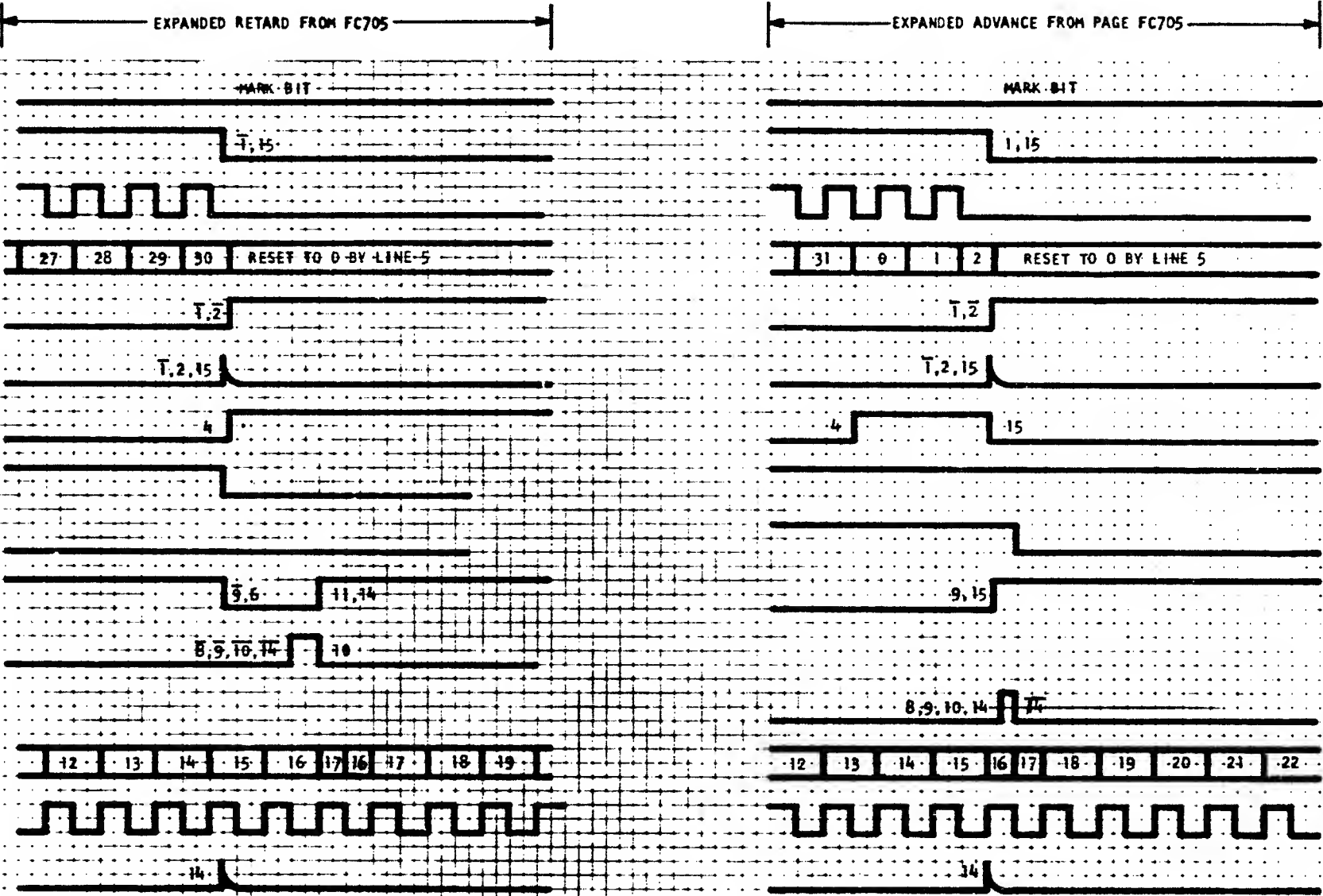
DATE	EC NUMBER	DATE	EC NUMBER	SCA XID START READ TIMING STR-			
SEP 66	419632			2 OR 4 WIRE - SEE NOTE 1			
DEC 66	419644			DATE		P-N	2231303
22APR68	419675					TYPE	1130
				IBM		FC704	

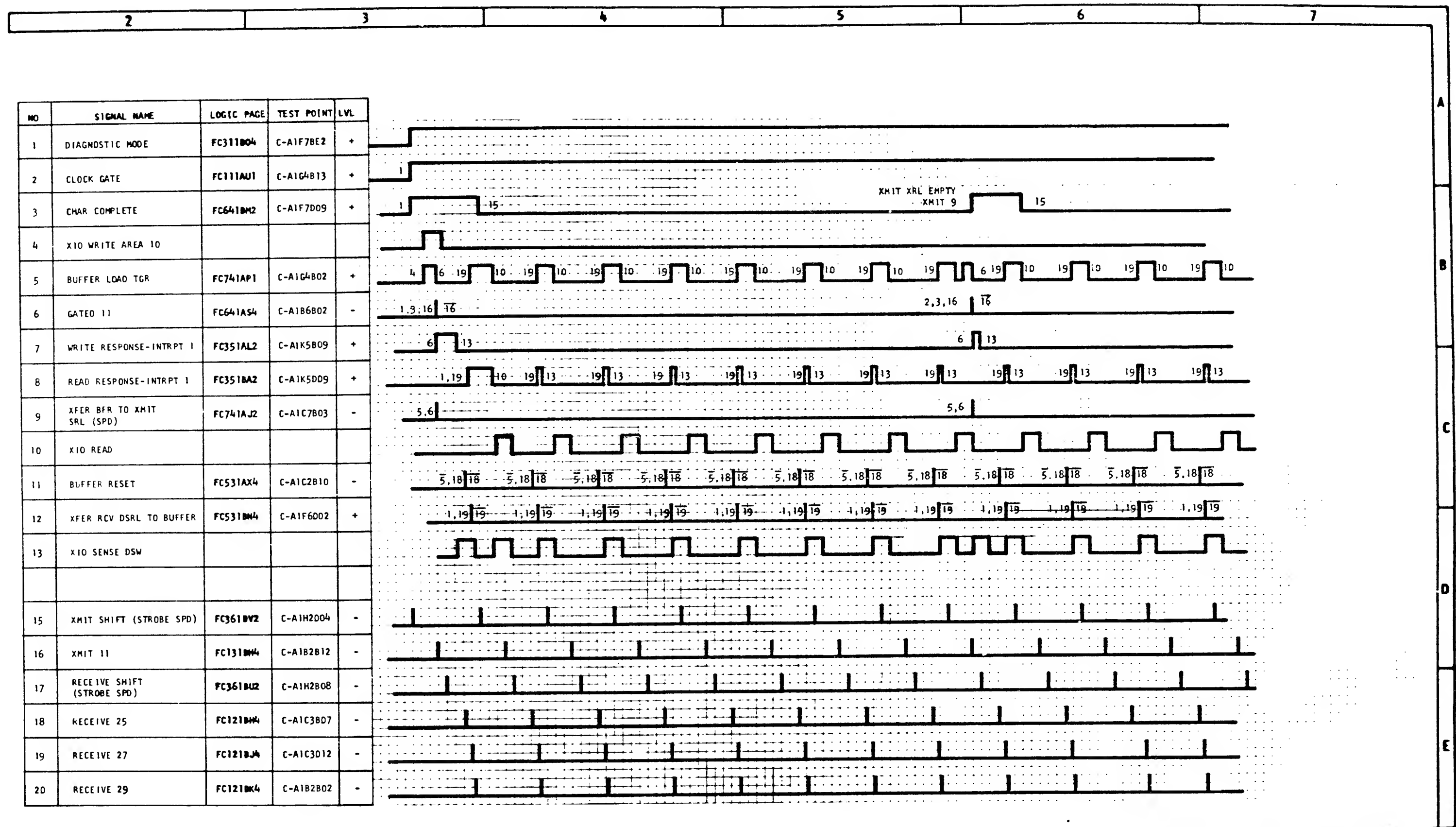


NOTES
1 VALUES OF 32 STEP COUNTER ARE SHOWN AT END OF PHASE COUNTER STEPPING
FOR VALUES OF PHASE COUNTER SHOWN GREATER THAN 31, THE ACTUAL P.C. COUNT IS VALUE MINUS 32.
2 P.C. REFERS TO PHASE COUNT. THE MEMORY F.F. IS TURNED ON AS THE PHASE COUNT REACHES 32.
* 17 STEP PHASE COUNTER PULSES
* 15 STEP PHASE COUNTER PULSES

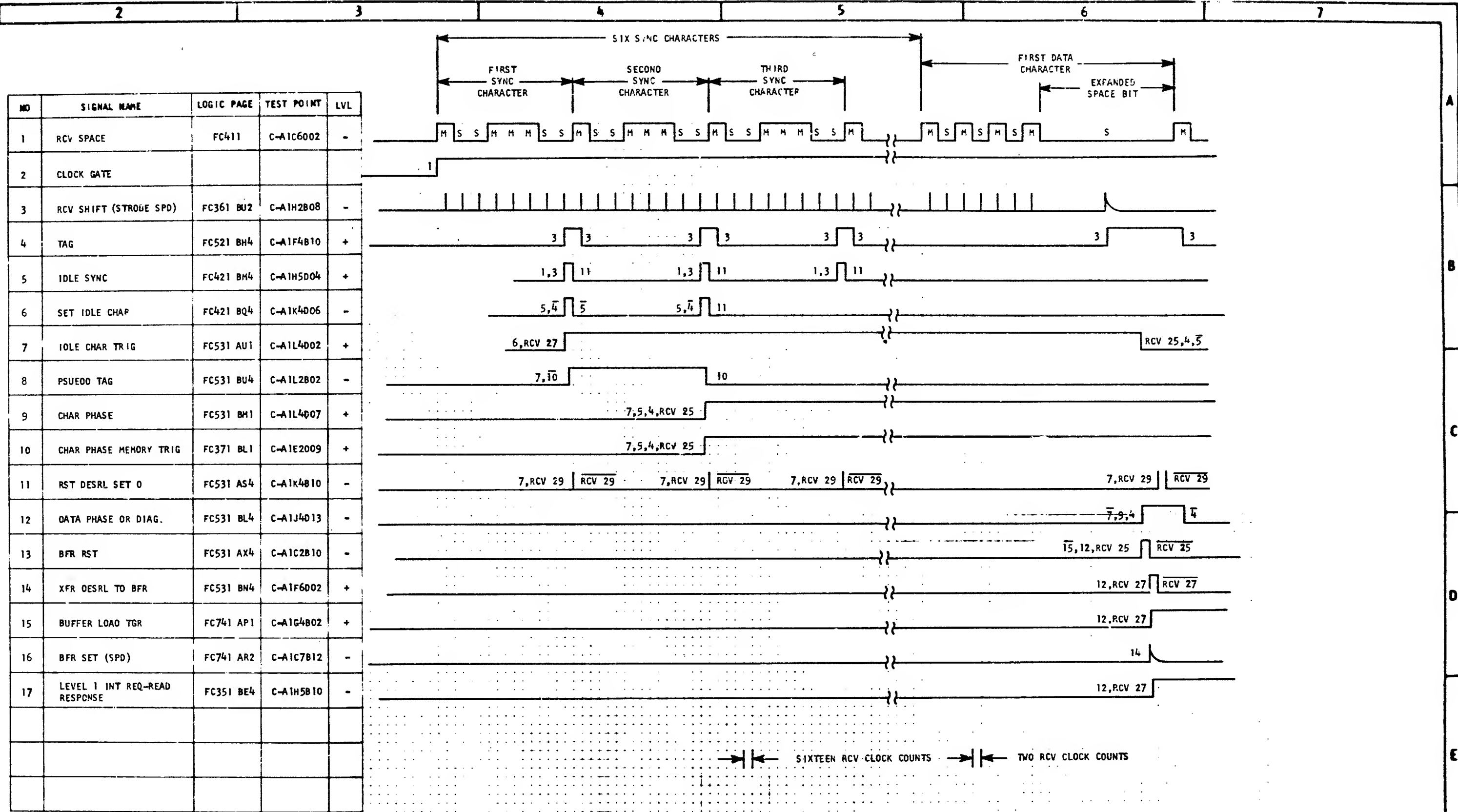
DATE	EC NUMBER	DATE	EC NUMBER	CAT PHASE COUNTER - STR		
SEP 66	419632			ADVANCE /RETARD REV CLOCK		
DEC 66	419644			DATE	AUG 66	P/M 2231304
22APR68	419675				TYPE	1130
				IBM FC705		

NO	SIGNAL NAME	LOGIC PAGE	TEST POINT	LVL
1	RCV SPACE	FC411BX4	C-A1J4D06	+
2	PHASE COUNTER CONTROL	FC211AE2	C-A1J2D09	+
3	STEP PHASE COUNTER	FC211AM4	C-A1D5B09	-
4	PHASE COUNTER VALUE	FC221		
5	RESET PHASE CTR.	FC211AM4	C-A1D5B08	-
6	END CORRECTION CYCLE SPD	FC211BH2	C-A1H2B03	-
7	MEMORY F.F.	FC231AS2	C-A1J2B09	+
8	AR1 F.F.	FC231AT2	C-A1J2B04	+
9	AR2 F.F.	FC231AW1	C-A1L3D07	+
10	AR3 F.F.	FC231AX1	C-A1L3D02	+
11	RETARD	FC231AZ4	C-A1H7D04	-
12	SET R1 (ADVANCE)	FC231AY4	C-A1J3R02	-
13	RCV CLOCK	FC121		
14	GATED OSC. A	FC101BG4	C-A1E5B04	-
15	RCV SHIFT (STROBE)	FC361BU2	C-A1H2B08	-

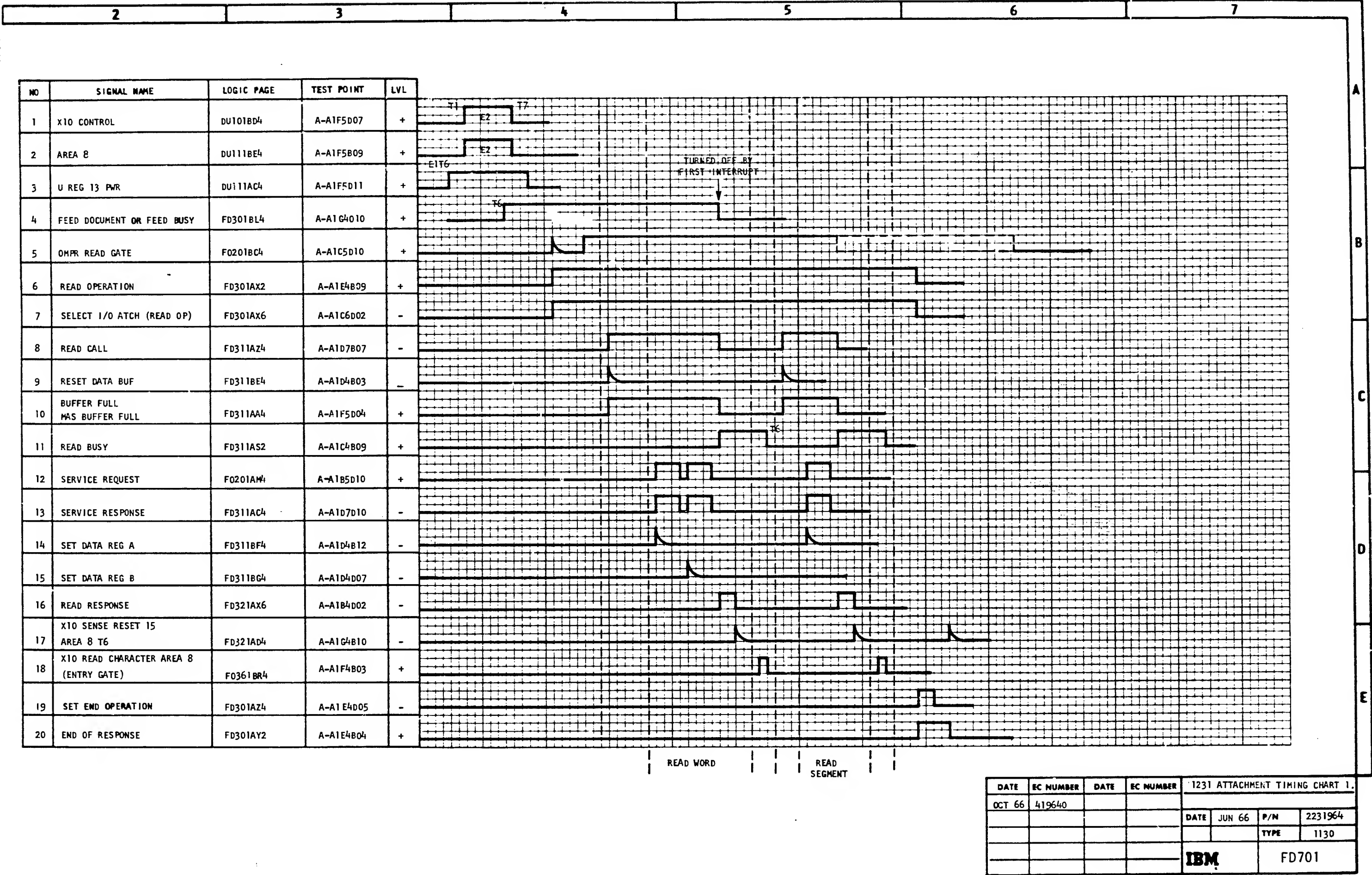


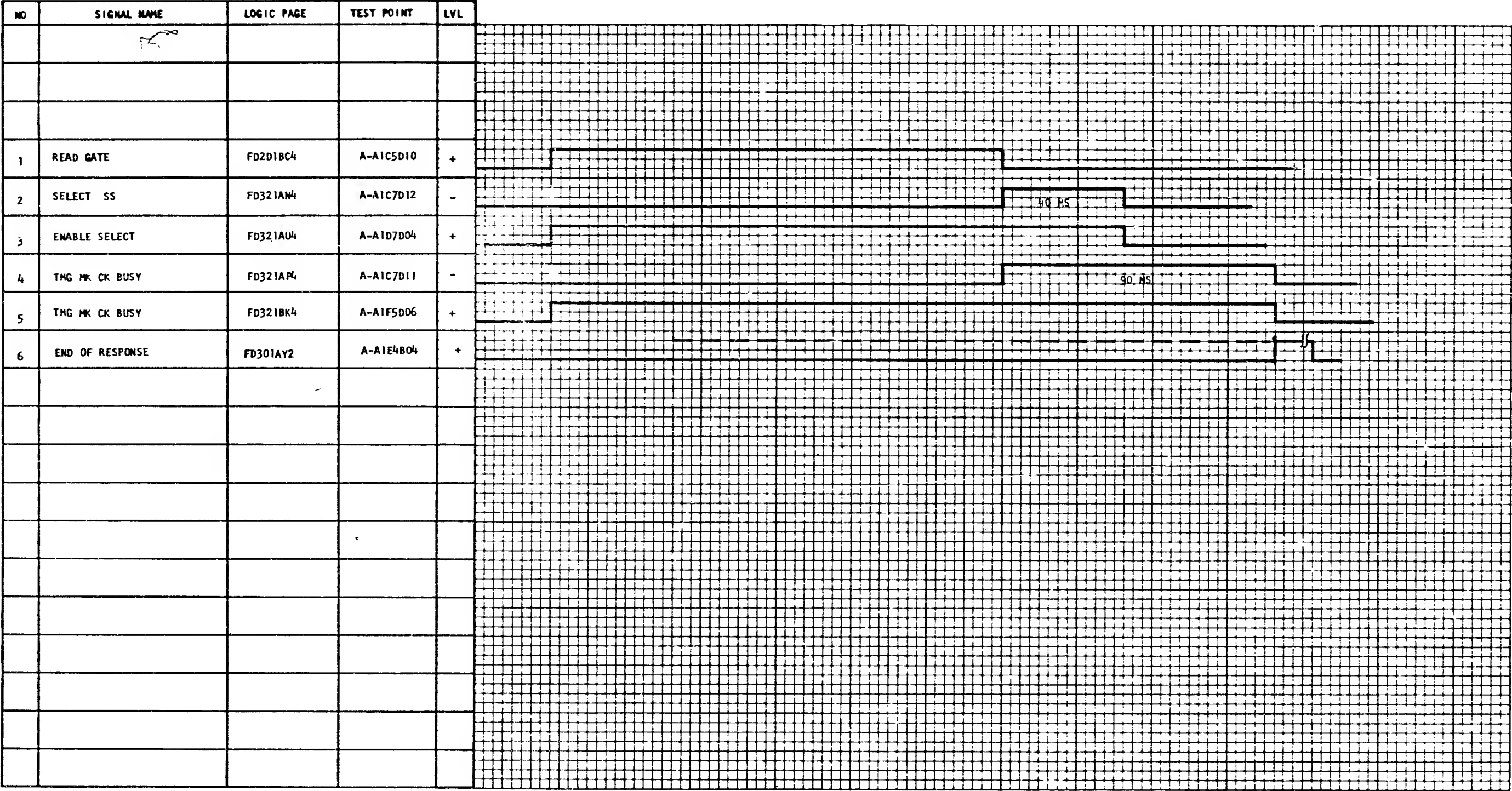


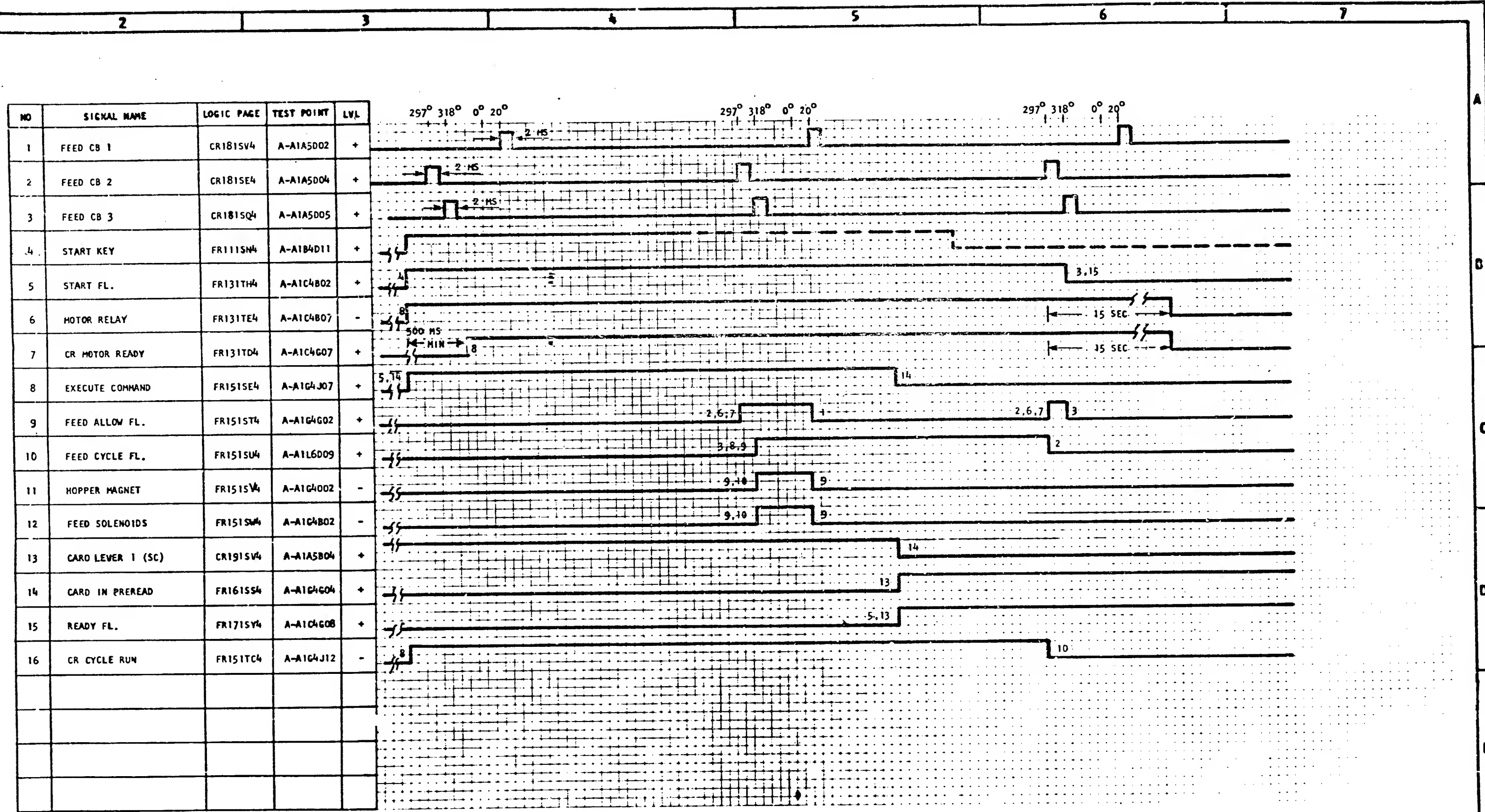
DATE	EC NUMBER	DATE	EC NUMBER	SCA DIAGNOSTIC MODE			
SEP 66	419632			STR TIMING			
DEC 66	419644			DATE	AUG 66	P/M	2231353
22APR68	419675					TYPE	1130
				IBM		FC707	



DATE	EC NUMBER	DATE	EC NUMBER	SCA X10 START READ TIMING BSC -			
22APR68	419675			2 OR 4 WIRE			
				DATE		P M	2231977
						TYPE	1130
				13M		FC708	

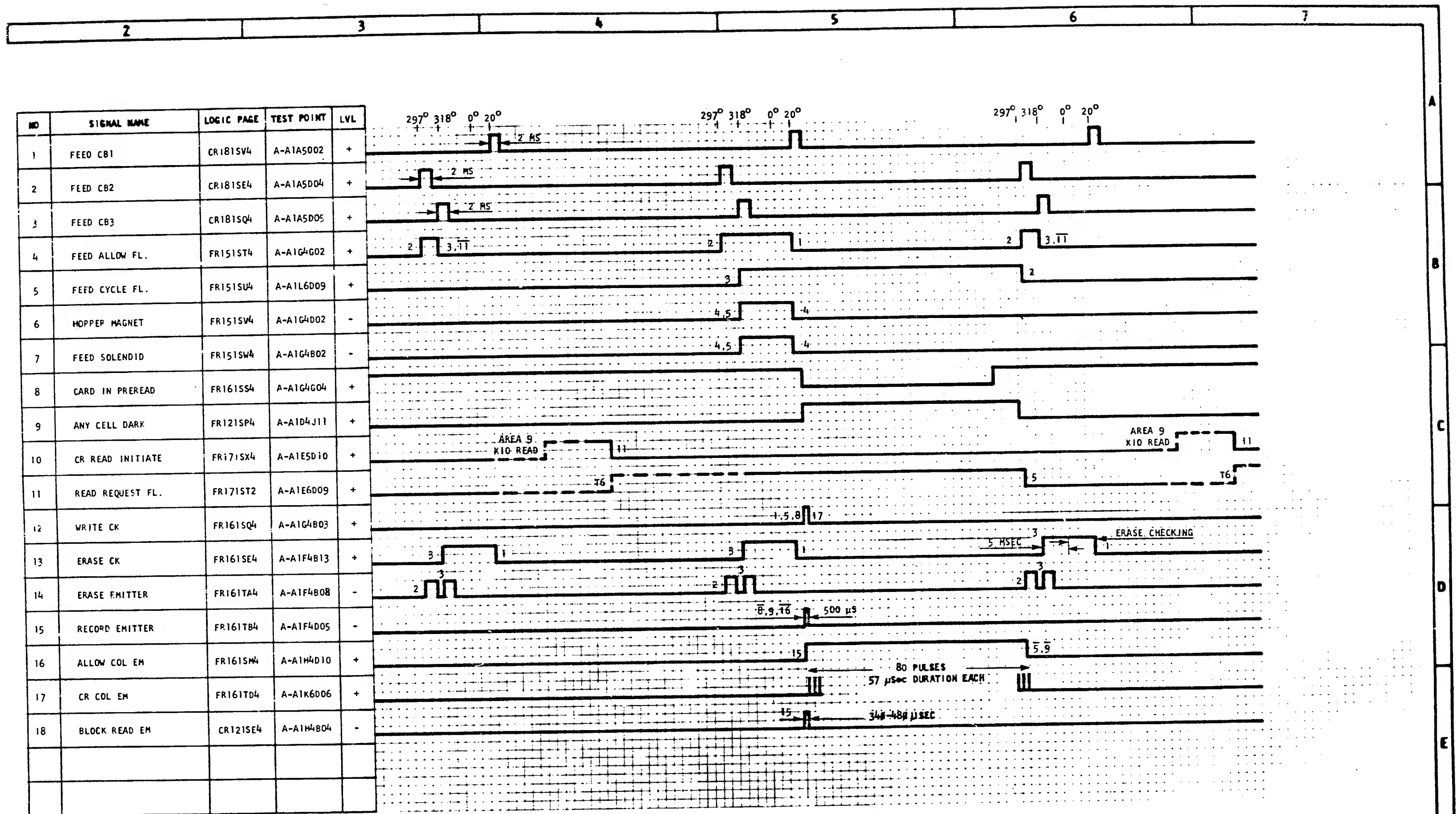






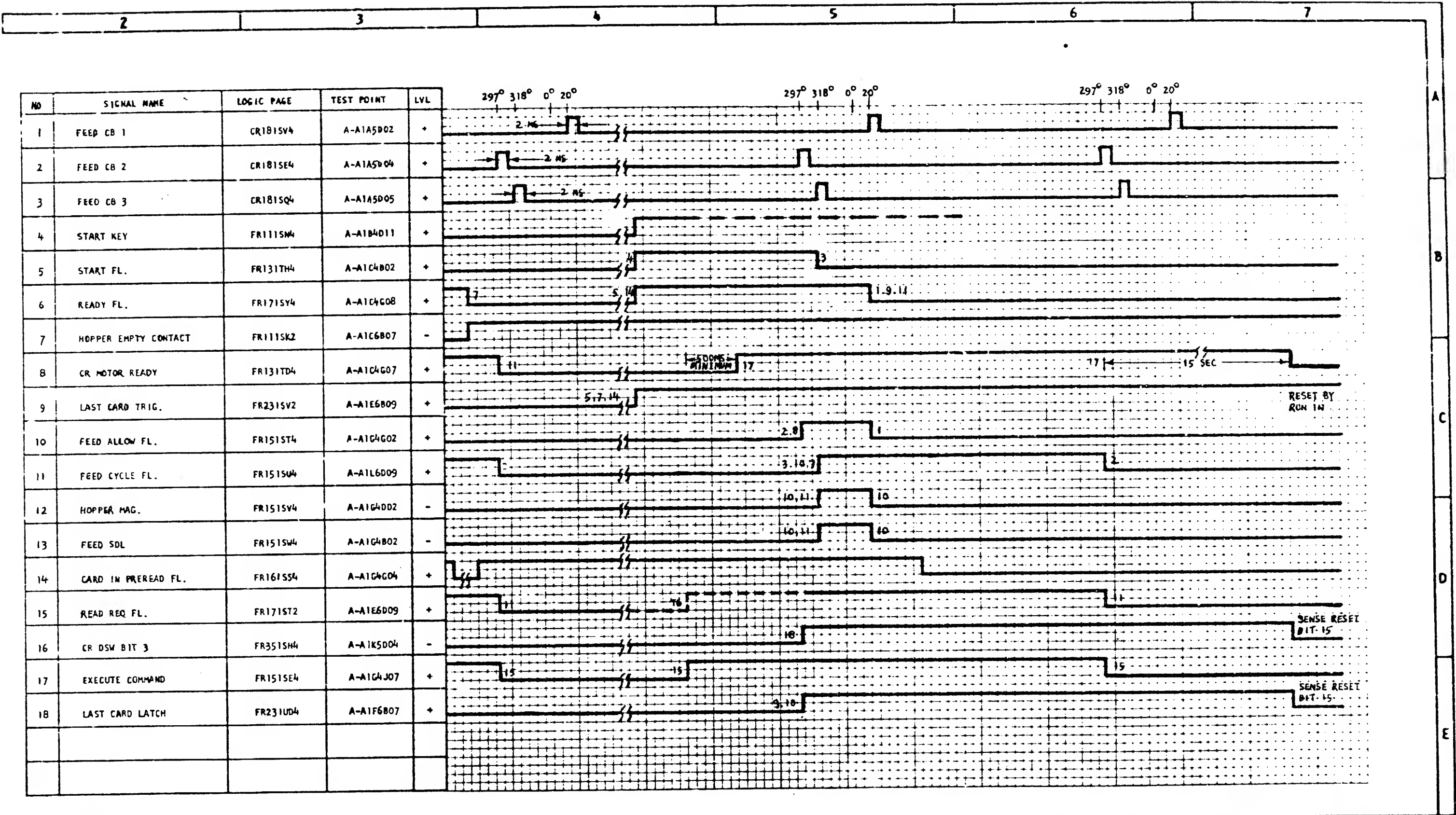
NOTES:
1. 360° = 60 MSEC

DATE	EC NUMBER	DATE	EC NUMBER	2501 CARD READER			
				RUN IN			
SEP 66	419632			DATE	SEP 66	P/M	2231337
						TYPE	1130
				IBM		FR701	



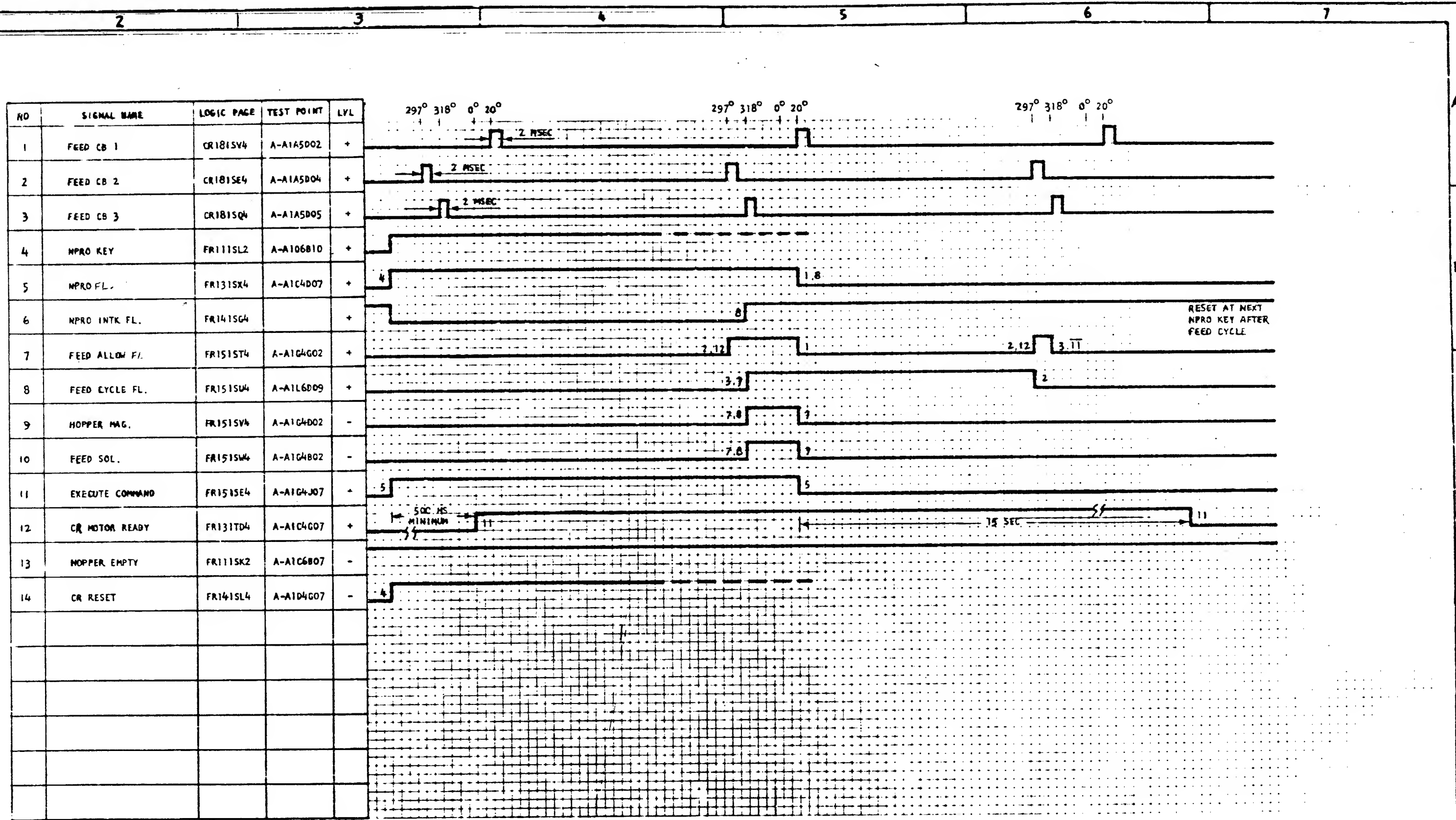
NOTES:
1. 360° = 60 MSEC

DATE	EC NUMBER	DATE	EC NUMBER	2501 CARD READER			
SEP 66	419632			READ CARD			
MAY67	419652			DATE	SEP 66	P/M	2231338
22APR68	419675					TYPE	1130
				IBM		FR711	



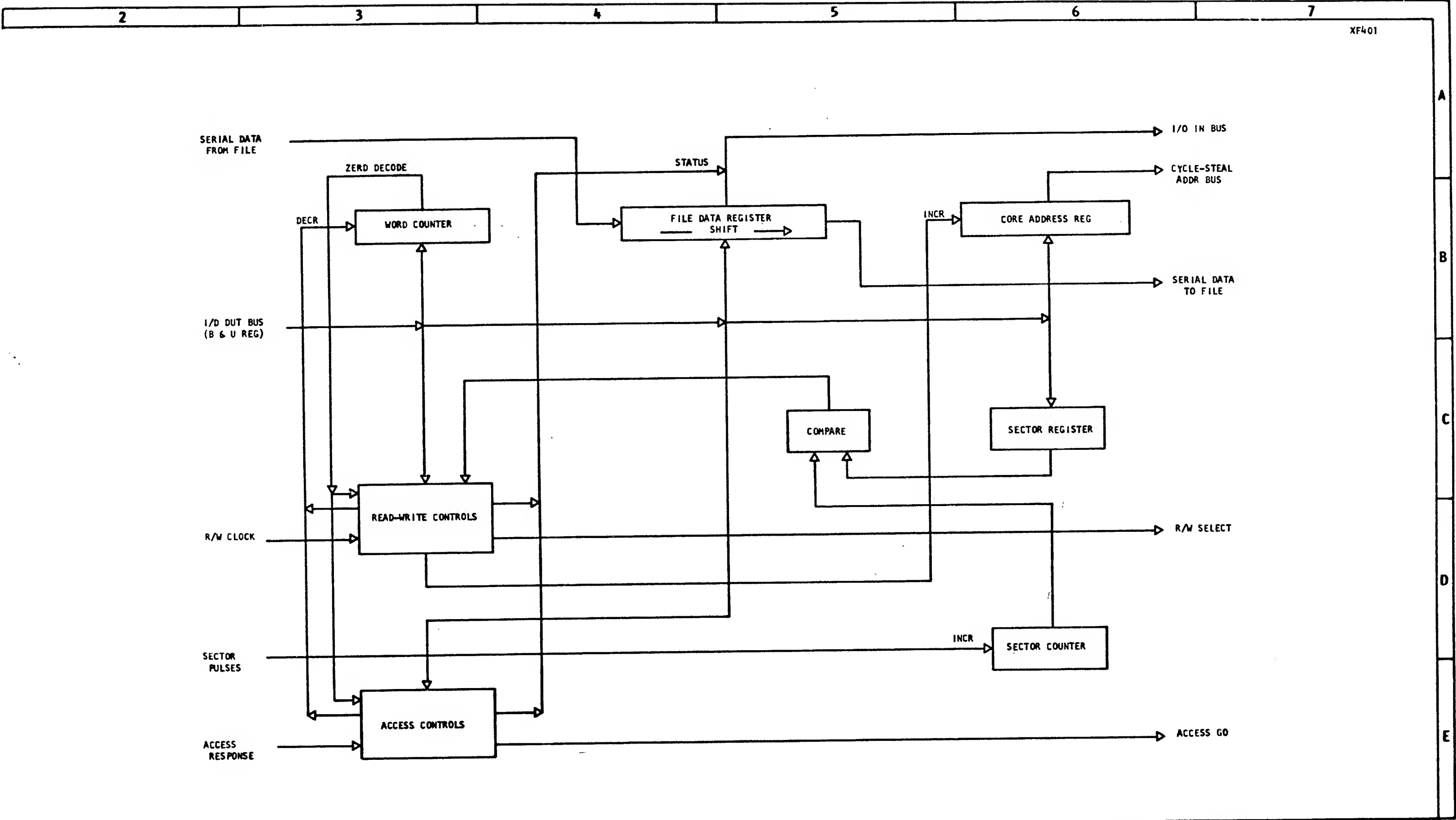
NOTES:
1. 360° = 60 MSEC

DATE	EC NUMBER	DATE	EC NUMBER	2501 CARD READER			
SEP 66	419632			LAST CARD			
				DATE	SEP 66	P/M	2231340
						TYPE	1130
				IBM		FR731	



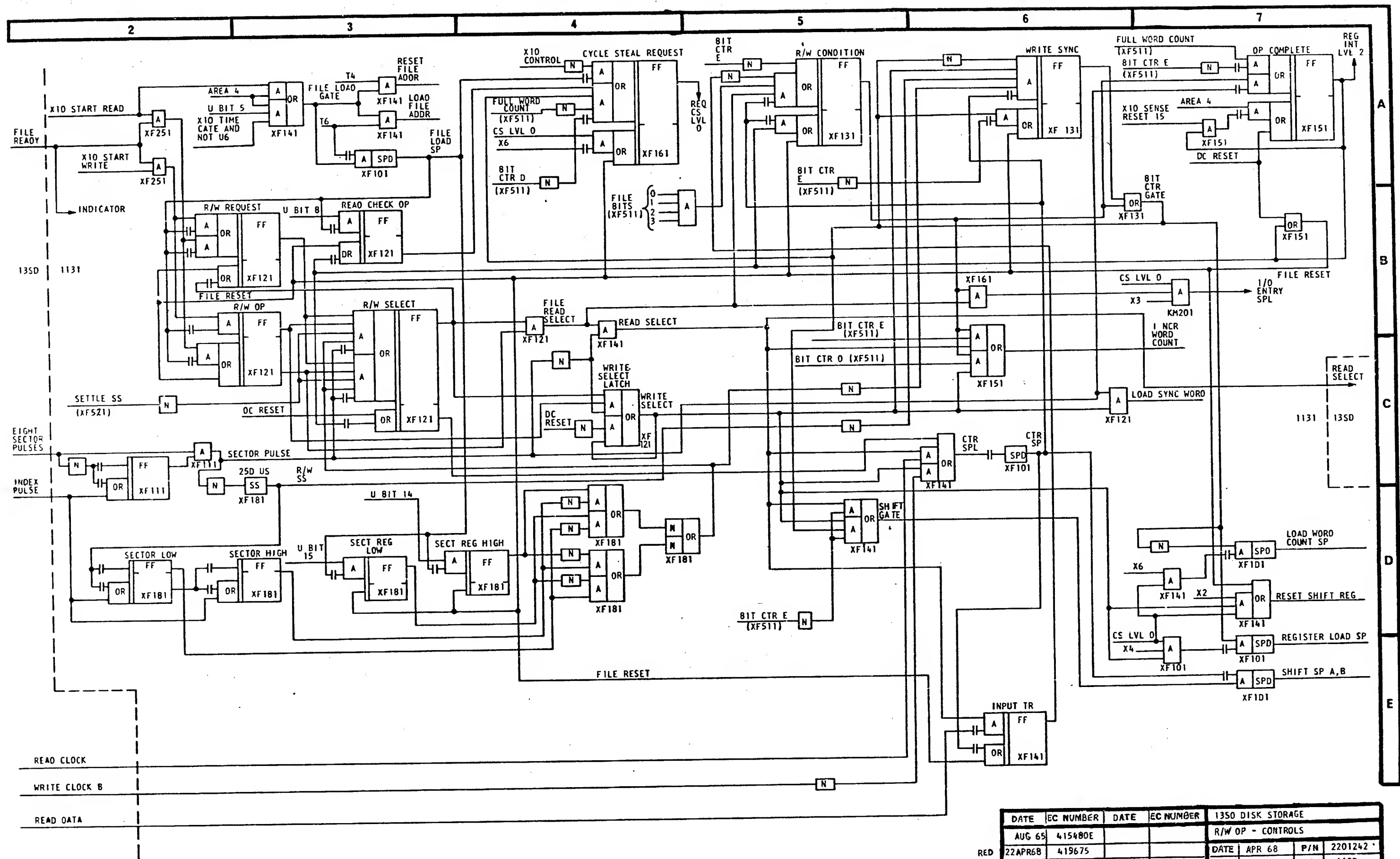
NOTES:
1. 360° = 60 MSEC

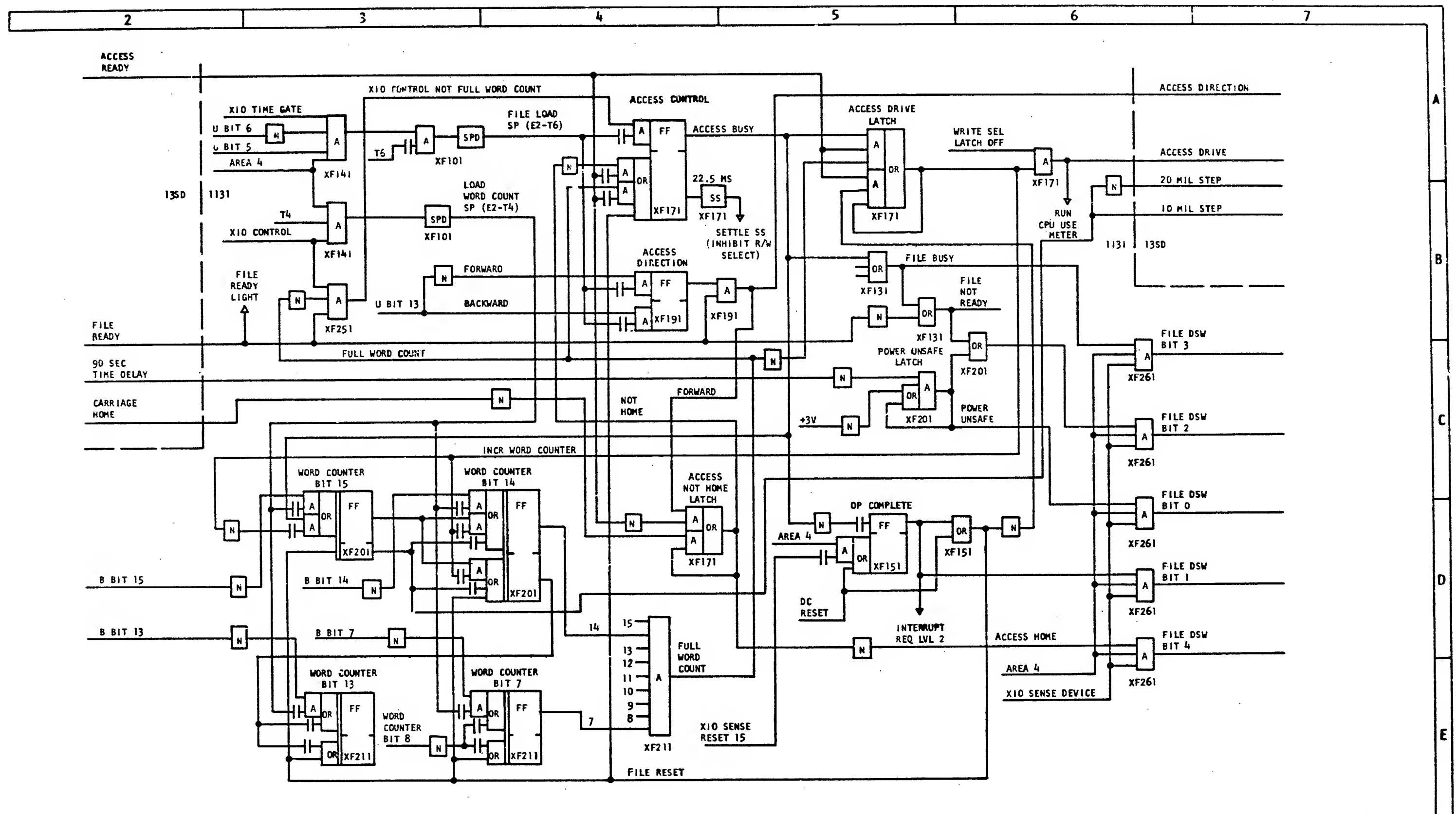
DATE	DC NUMBER	DATE	DC NUMBER	2501 CARD READER NPRO			
SEP 66	419632			DATE	SEP 66	P/N	2231342
						TYPE	1130
				IBM		FR751	



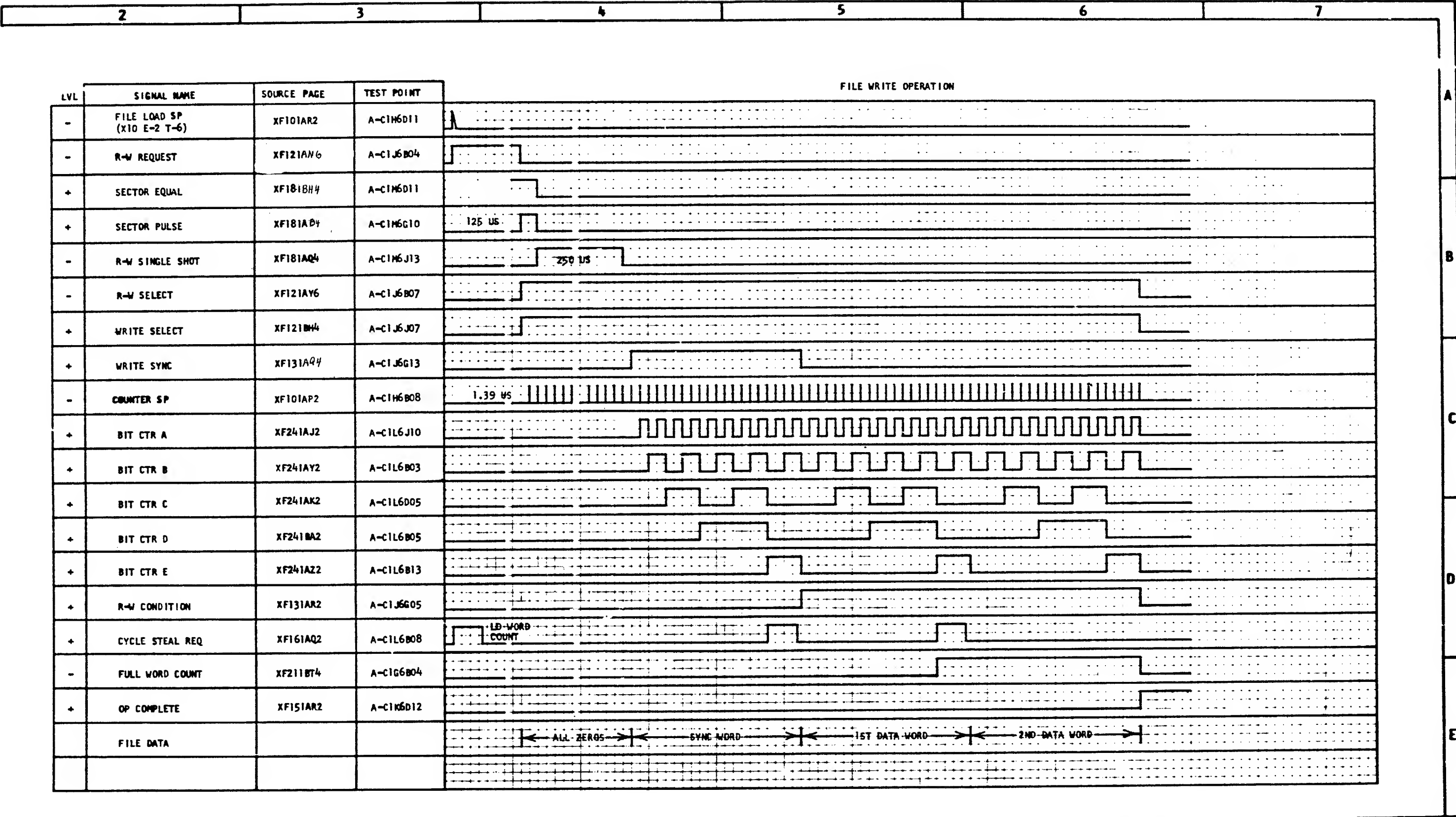
DATE	EC NUMBER	DATE	EC NUMBER	DISK FILE UNIT DATA			
AUG-65	41548D E			AND CONTRDL DIAGRAM			
				DATE		P/N	2201241
						TYPE	113D
				IBM		XF401	

XF401

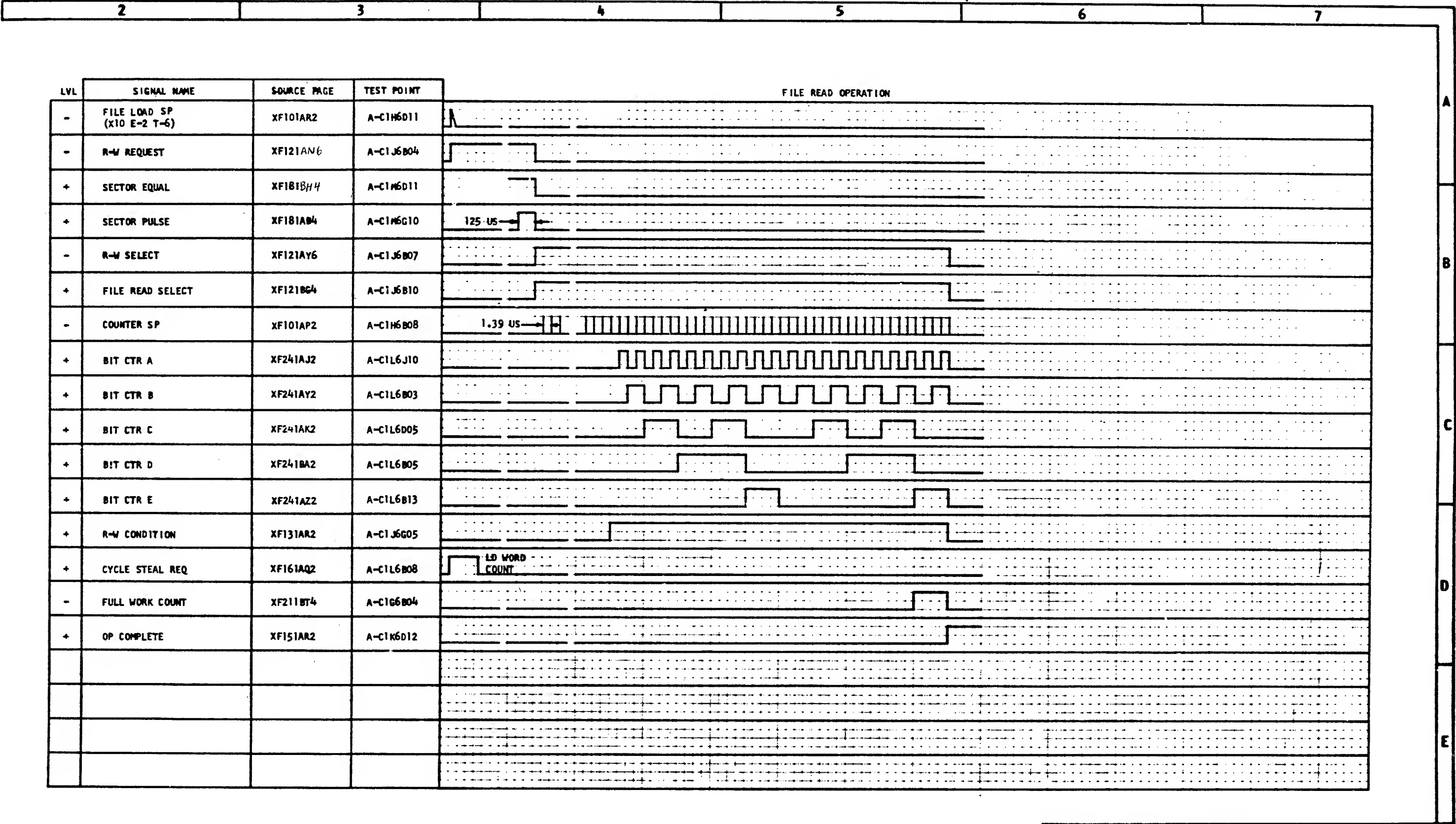




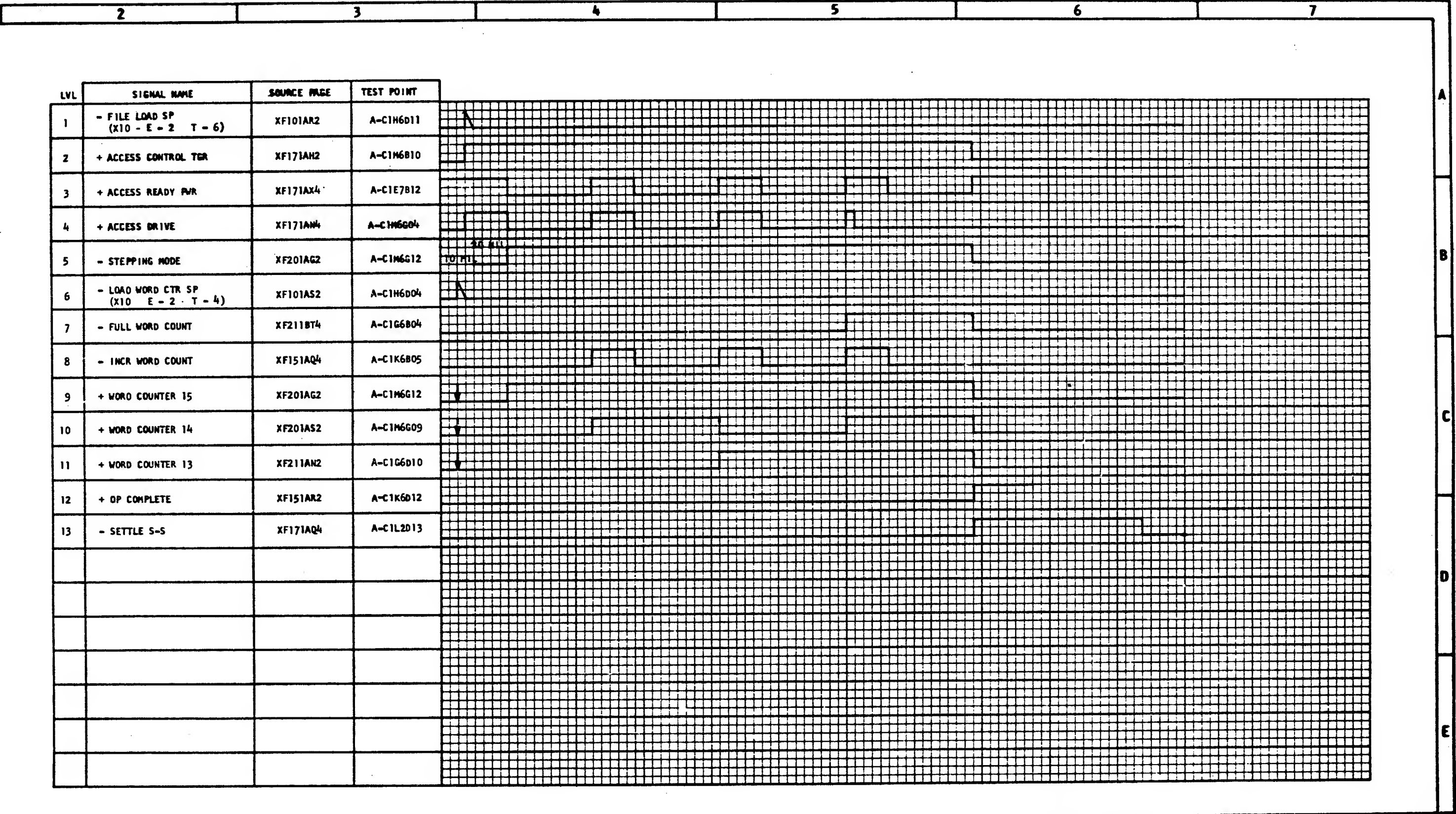
DATE	EC NUMBER	DATE	EC NUMBER	DISK FILE-CONTROL OP (ACCESS)			
AUG 65	415480E						
OCT 65	415483A			DATL	APR 68	P/N	2201244
22APR68	419675					TYPE	1130
RED				IBM		XF521	



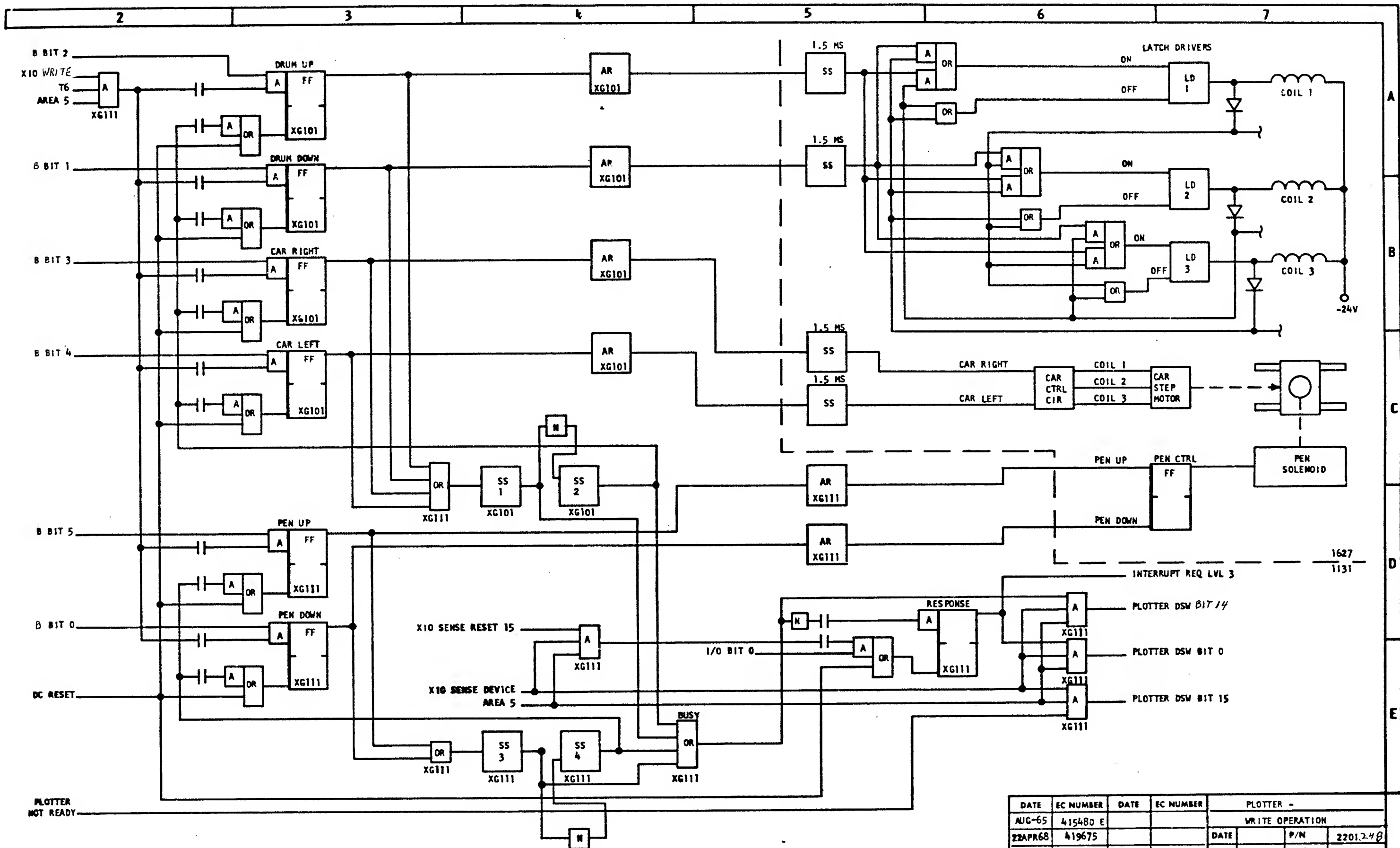
DATE	EC NUMBER	DATE	EC NUMBER	DISC FILE		
AUG-65	415480 E			WRITE TIMING		
22APR68	419675			DATE	3-18-65	P/M 2201245
					TYPE	1130
				IBM		XF701

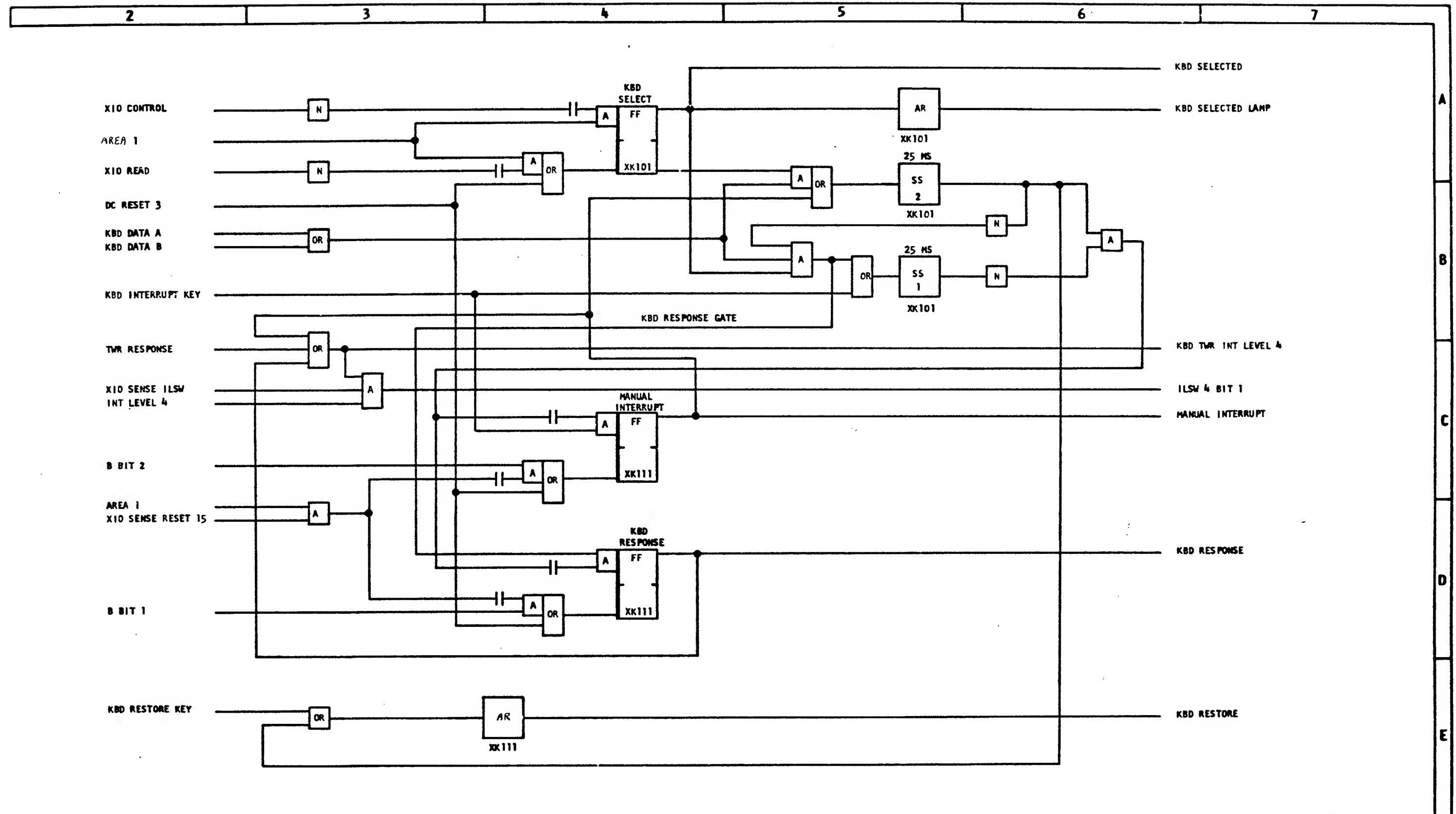


DATE	EC NUMBER	DATE	EC NUMBER	DISC FILE		
AJG-65	415480 E			READ TIMING		
22APR68	419675			DATE	3-17-65	P/N 2201246
					TYPE	1130
				IBM		XF711

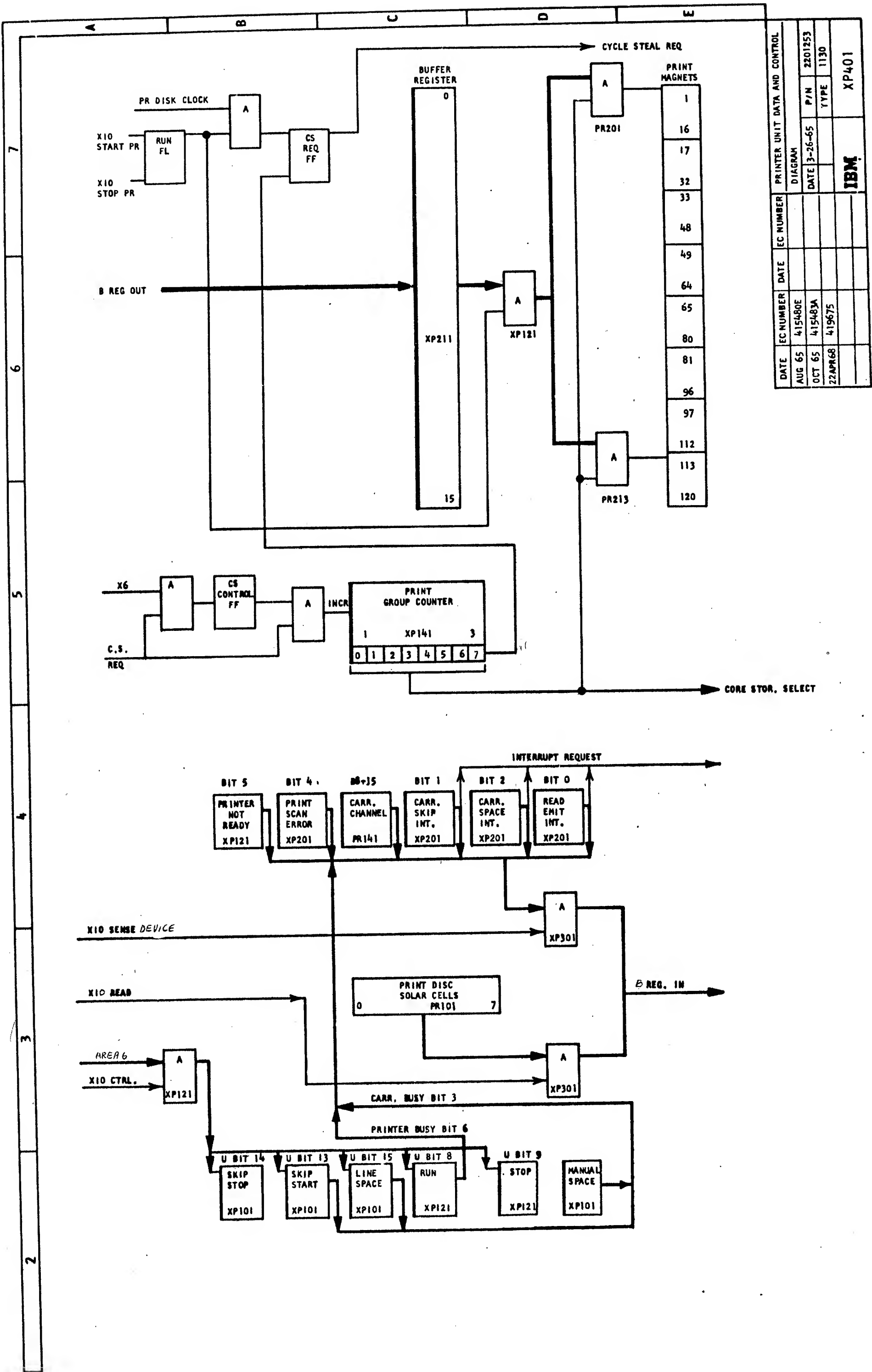


DATE	EC NUMBER	DATE	EC NUMBER	DISK FILE -		
AUG-65	415480 E			ACCESS TIMING		
22APR68	419675			DATE	3-19-65 P/M	2201247
					TYPE	1130
				IBM		XF721

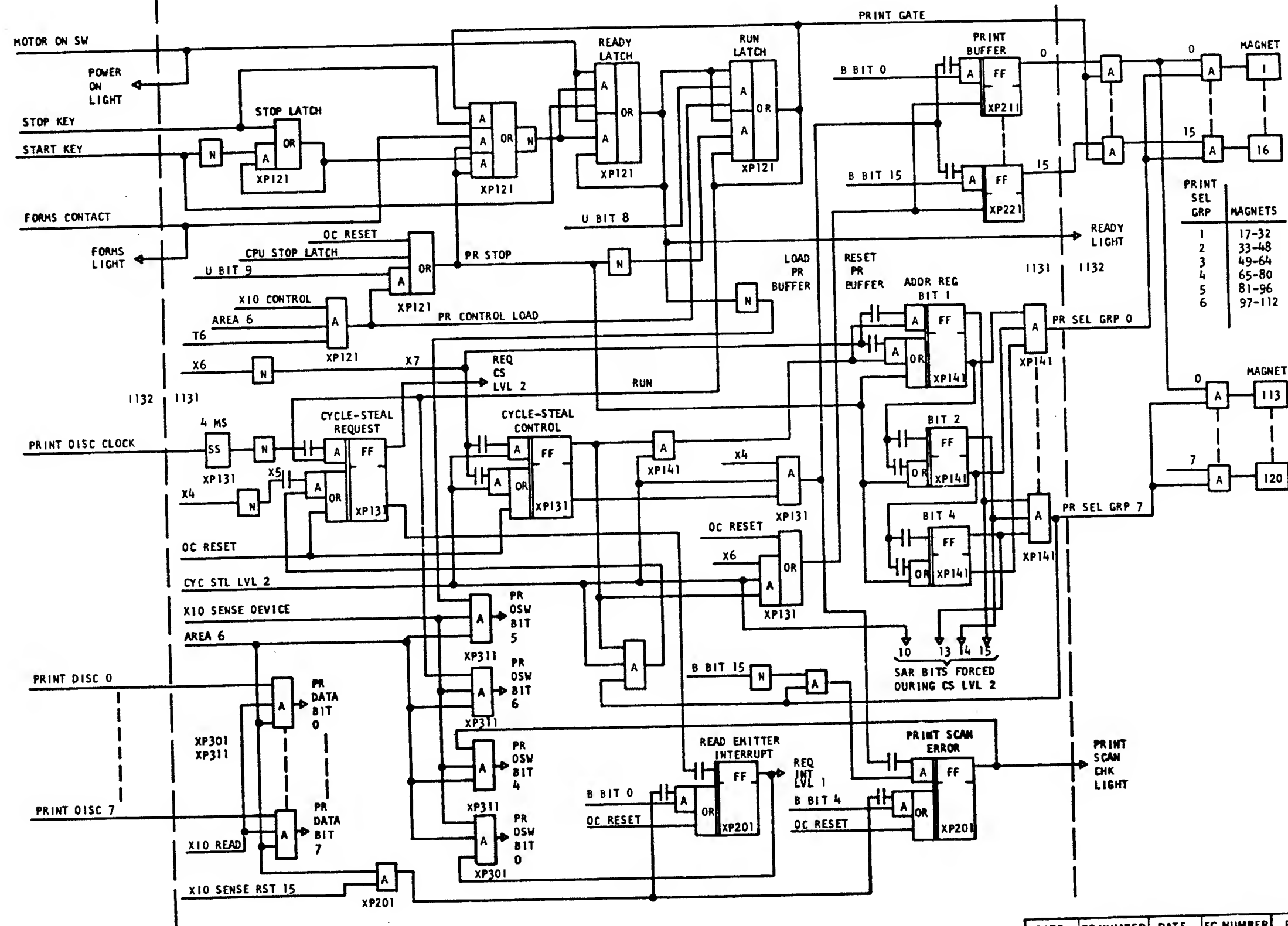




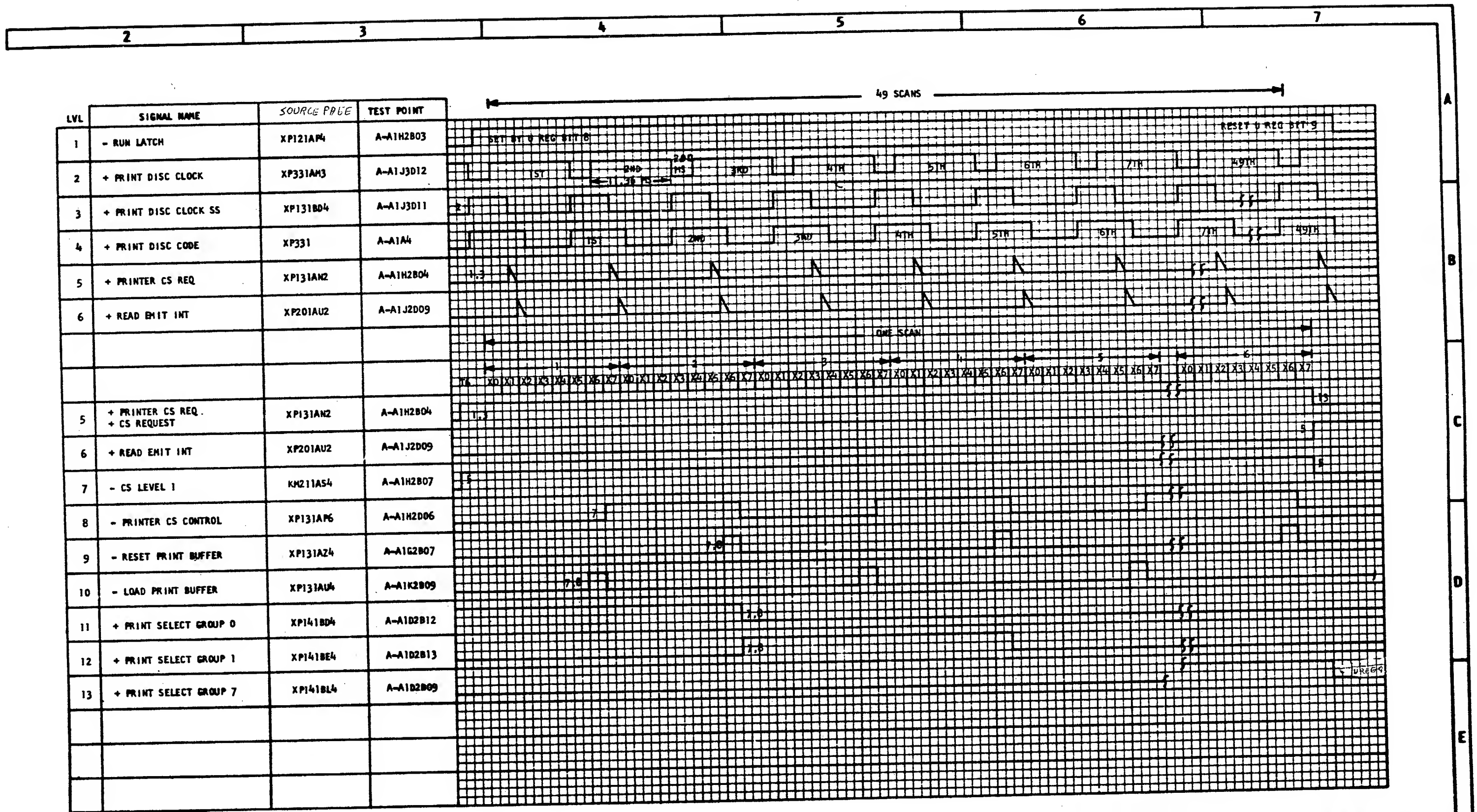
DATE	EC NUMBER	DATE	EC NUMBER	KEYBOARD READ 6			
AUG-65	415480 E			CONTROL OPS			
22APR68	419675			DATE	6-28-65	P/N	2201250
						TYPE	1130
				IBM		XK501	



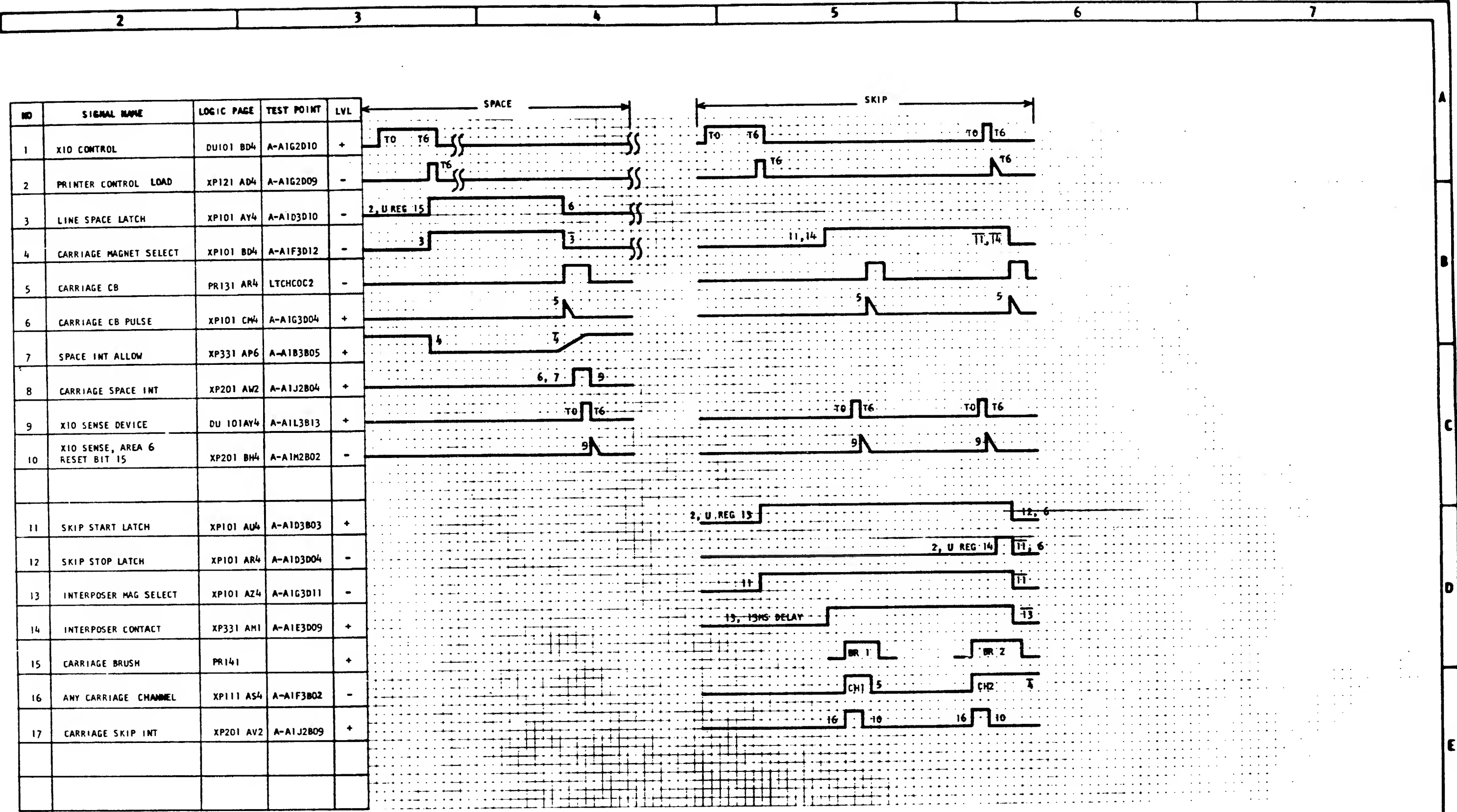
PRINTER UNIT DATA AND CONTROL			
DIAGRAM			
DATE	EC NUMBER	DATE	P/N
AUG 65	415480E		2201253
OCT 65	415483A		
22APR68	419675		
		TYPE	XP401
		IBM	



DATE	EC NUMBER	DATE	EC NUMBER	PRINTER WRITE OP			
AUG 65	415480E			(READ EMITTER, PRINT)			
22APR68	419675			DATE	APR 68	P/N	2201254
						TYPE	1130
				IBM		XP501	

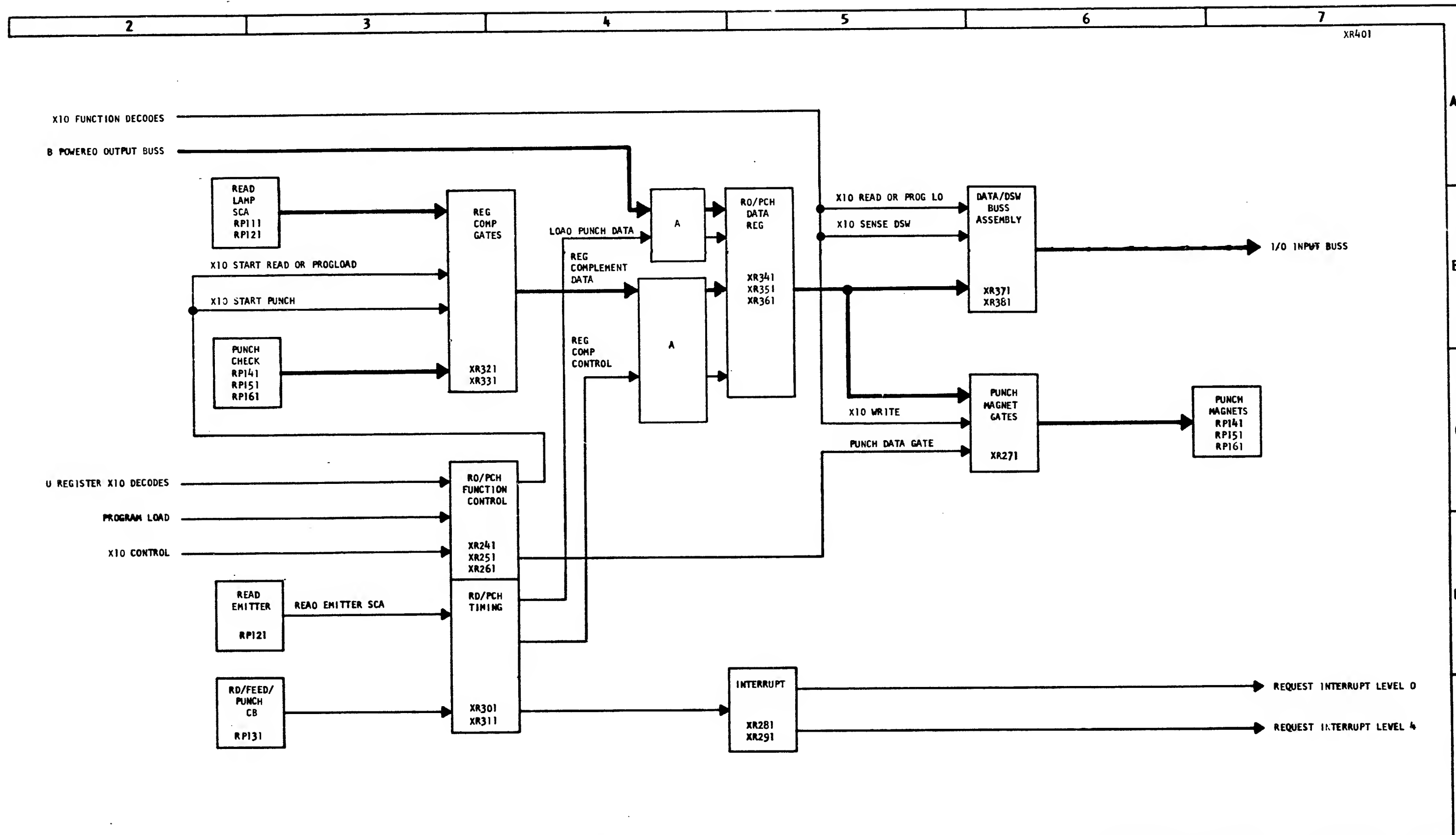


DATE	SC NUMBER	DATE	SC NUMBER	PRINTER WRITE TIMING		
AUG-65	415480 E			(READ EMITTER, PRINT)		
22APR68	419675			DATE	P/N	2201256
					TYPE	1130
				IBM XP701		

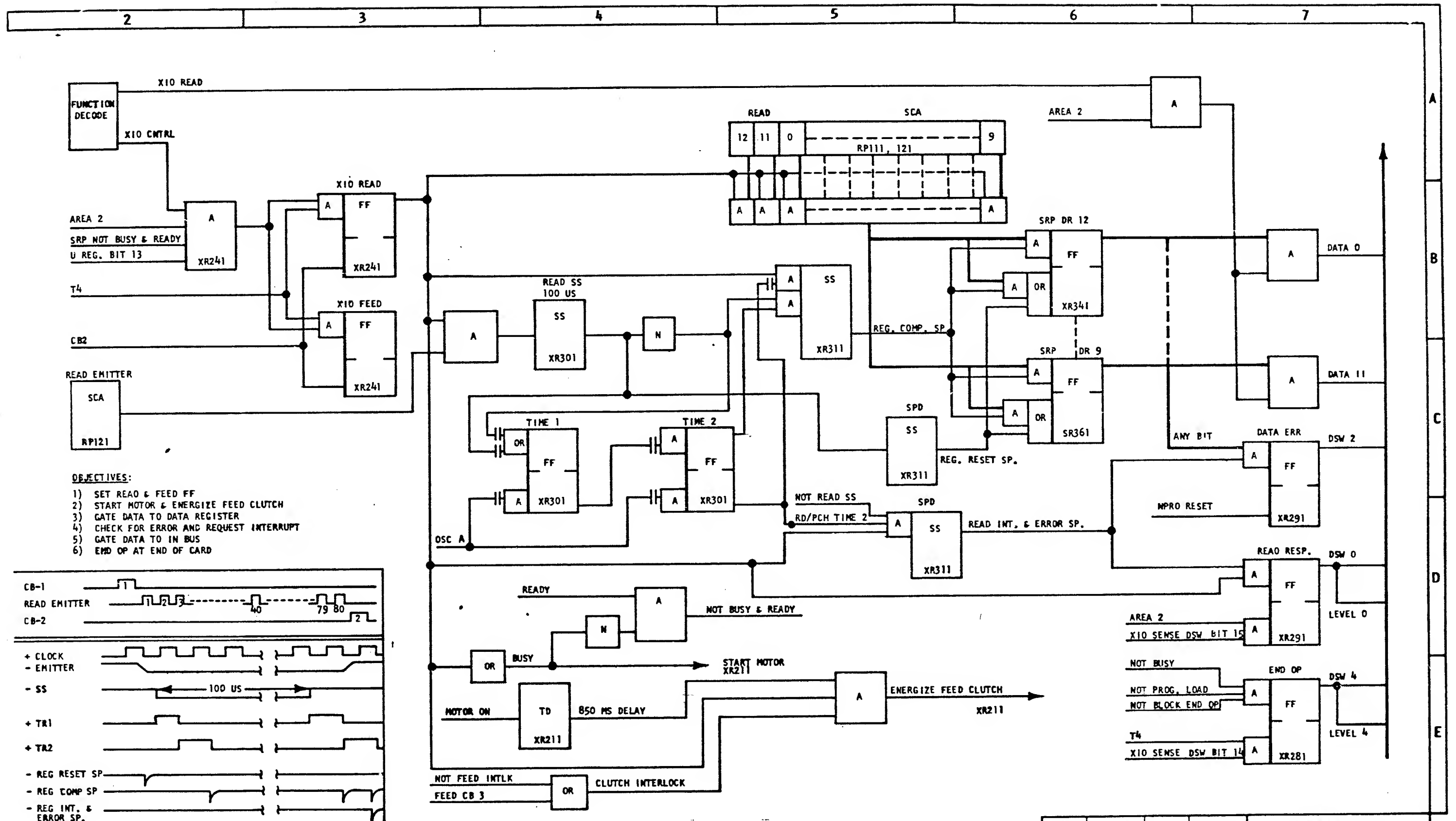


DATE	EC NUMBER	DATE	EC NUMBER	PRINTER CONTROL TIMING			
AUG 65	415480			START, STOP, SPACE			
AUG 66	419622			DATE		P/M	2201257
22APR68	419675					TIME	1130
				IBM		XP711	

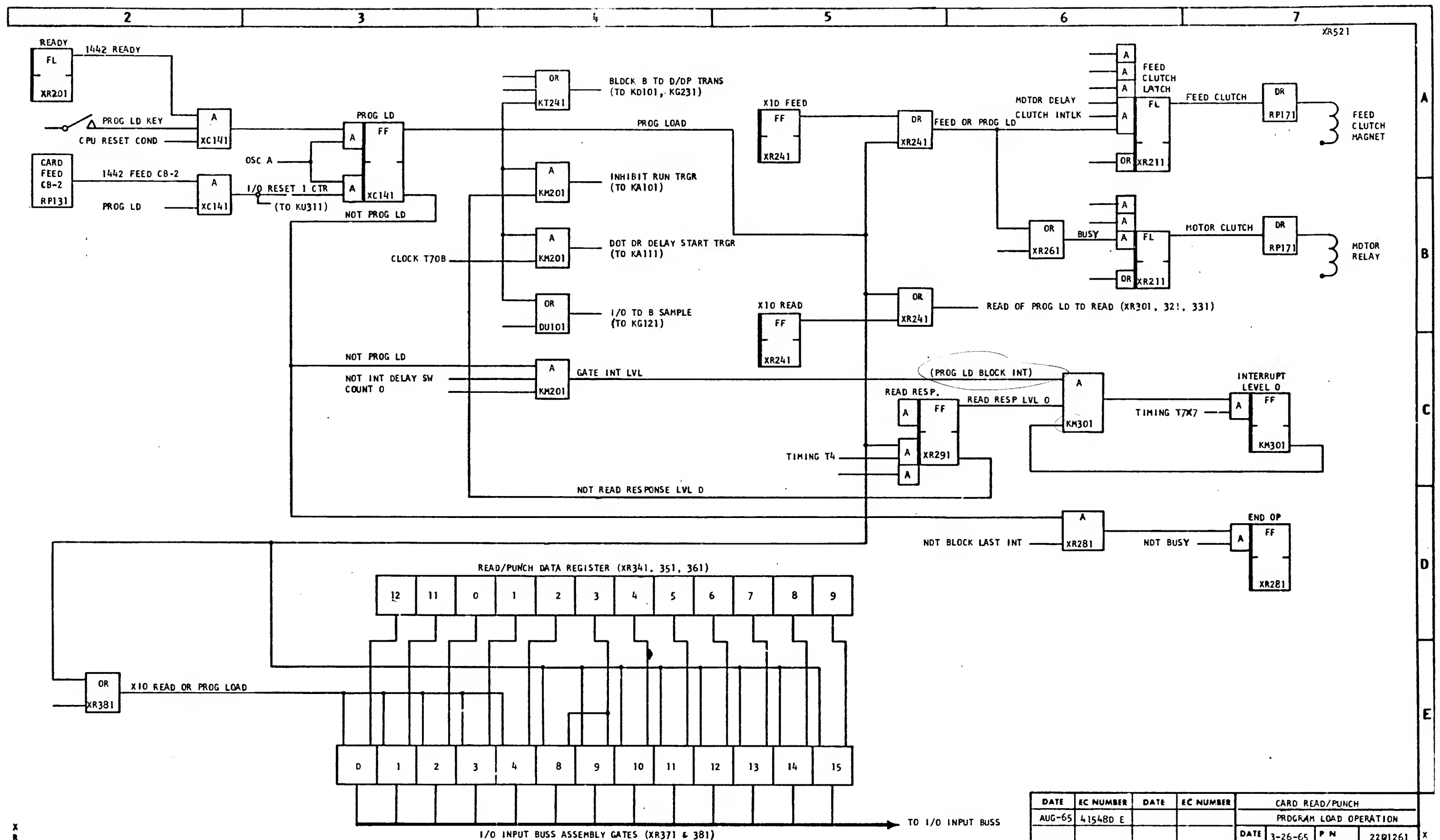
X
P
7
1
1



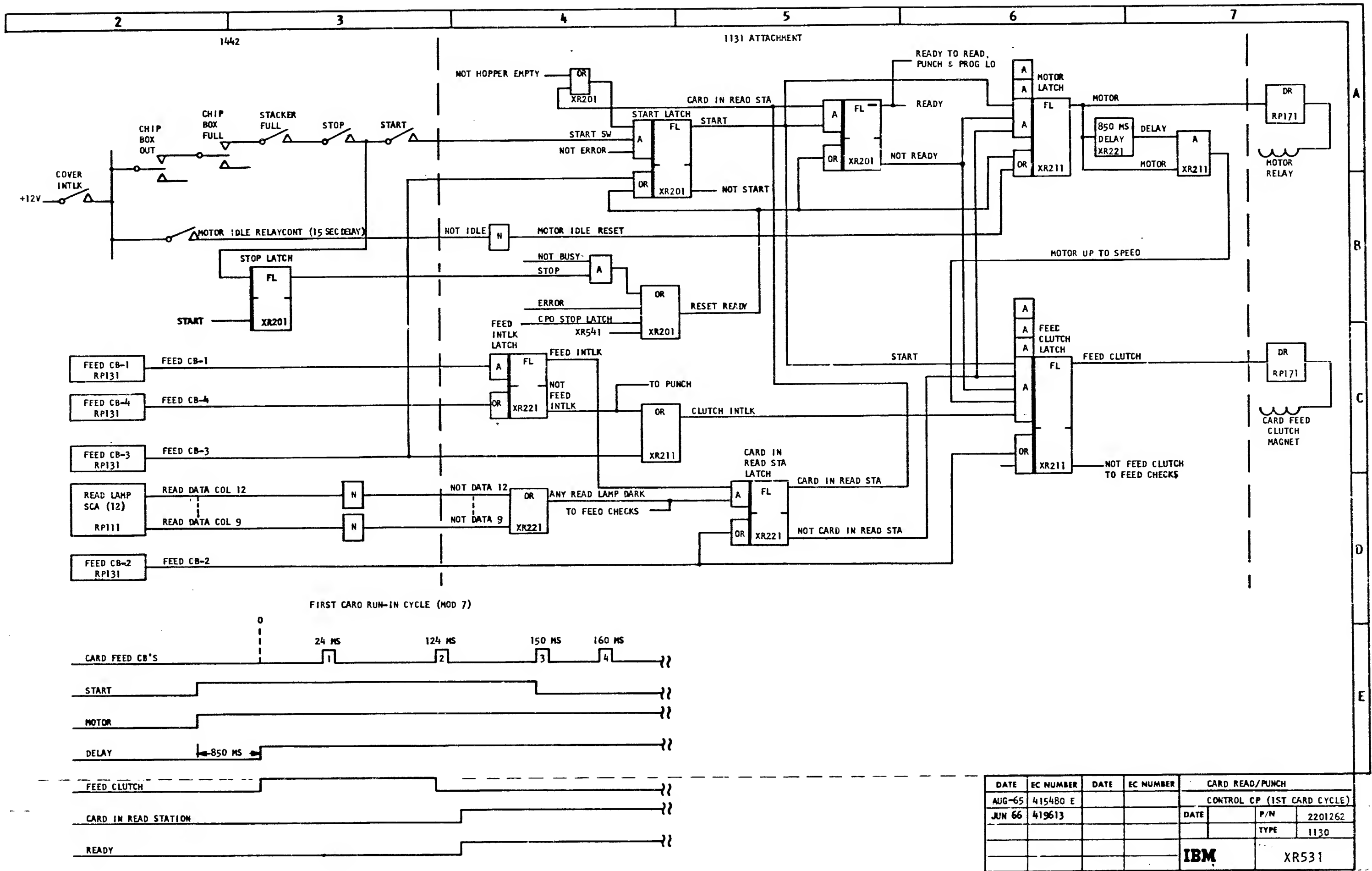
DATE	EC NUMBER	DATE	EC NUMBER	CARD READ/PUNCH UNIT DATA			
AUG-65	415480 E			AND CONTROL DIAGRAM			
				DATE	3-25-65	P/N	2201258
						TYPE	1130
				IBM		XR401	



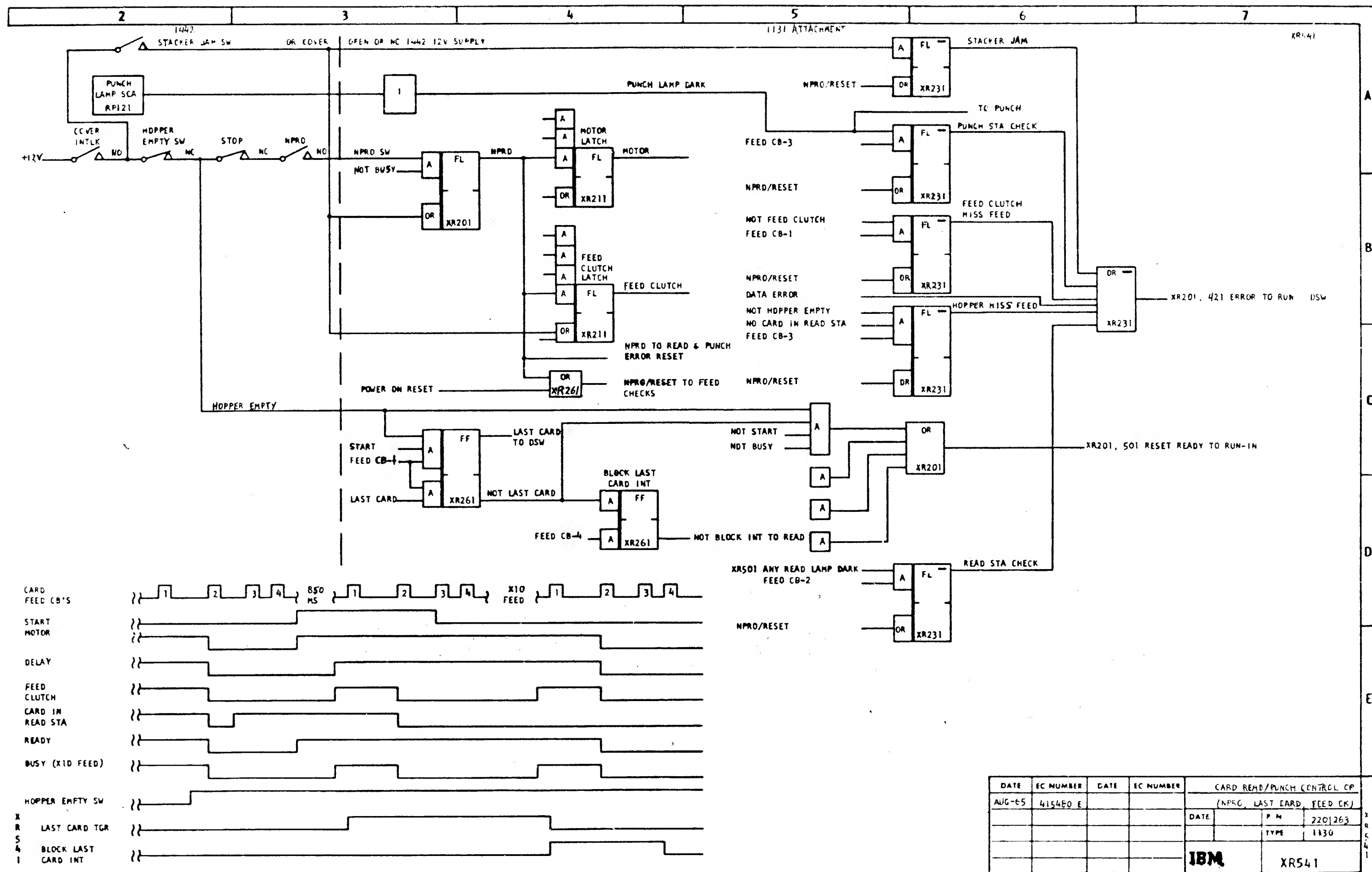
DATE	EC NUMBER	DATE	EC NUMBER	CARD READ/PUNCH -	
AUG-65	415480 E			READ OPERATION	
OCT 65	415483A			DATE 3-25-65	P/N 2201260
22APR68	419675			TYPE	113D
				IBM	
				XR511	

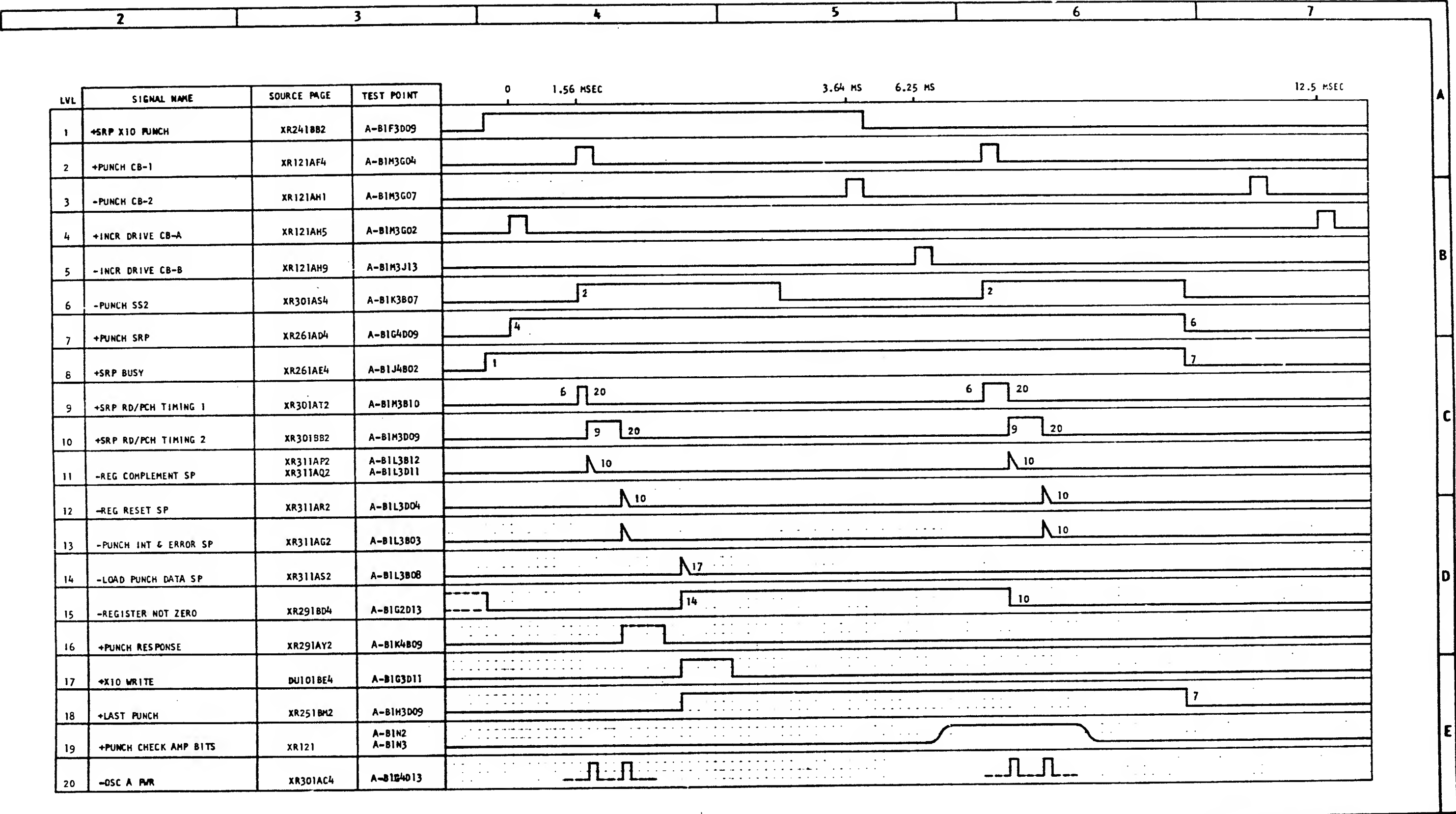


DATE	EC NUMBER	DATE	EC NUMBER	CARD READ/PUNCH			
AUG-65	41548D E			PROGRAM LOAD OPERATION			
		DATE	3-26-65	P N	2201261		
				TYPE	1130		
				IBM		XR521	

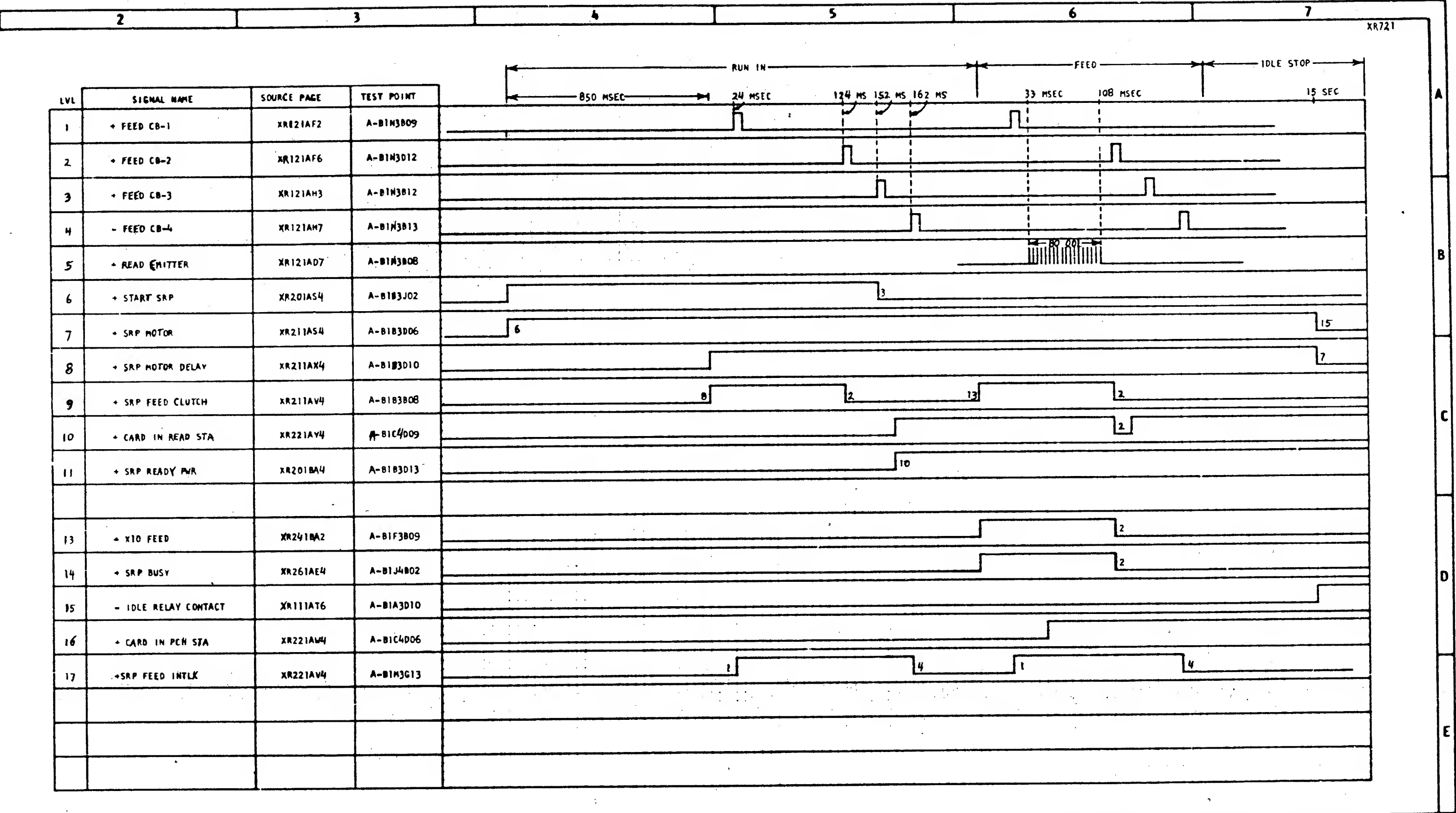


DATE	EC NUMBER	DATE	EC NUMBER	CARD READ/PUNCH		
AUG-65	415480 E			CONTROL CP (1ST CARD CYCLE)		
JUN 66	419613			DATE	P/N	2201262
					TYPE	1130
				IBM		
				XR531		

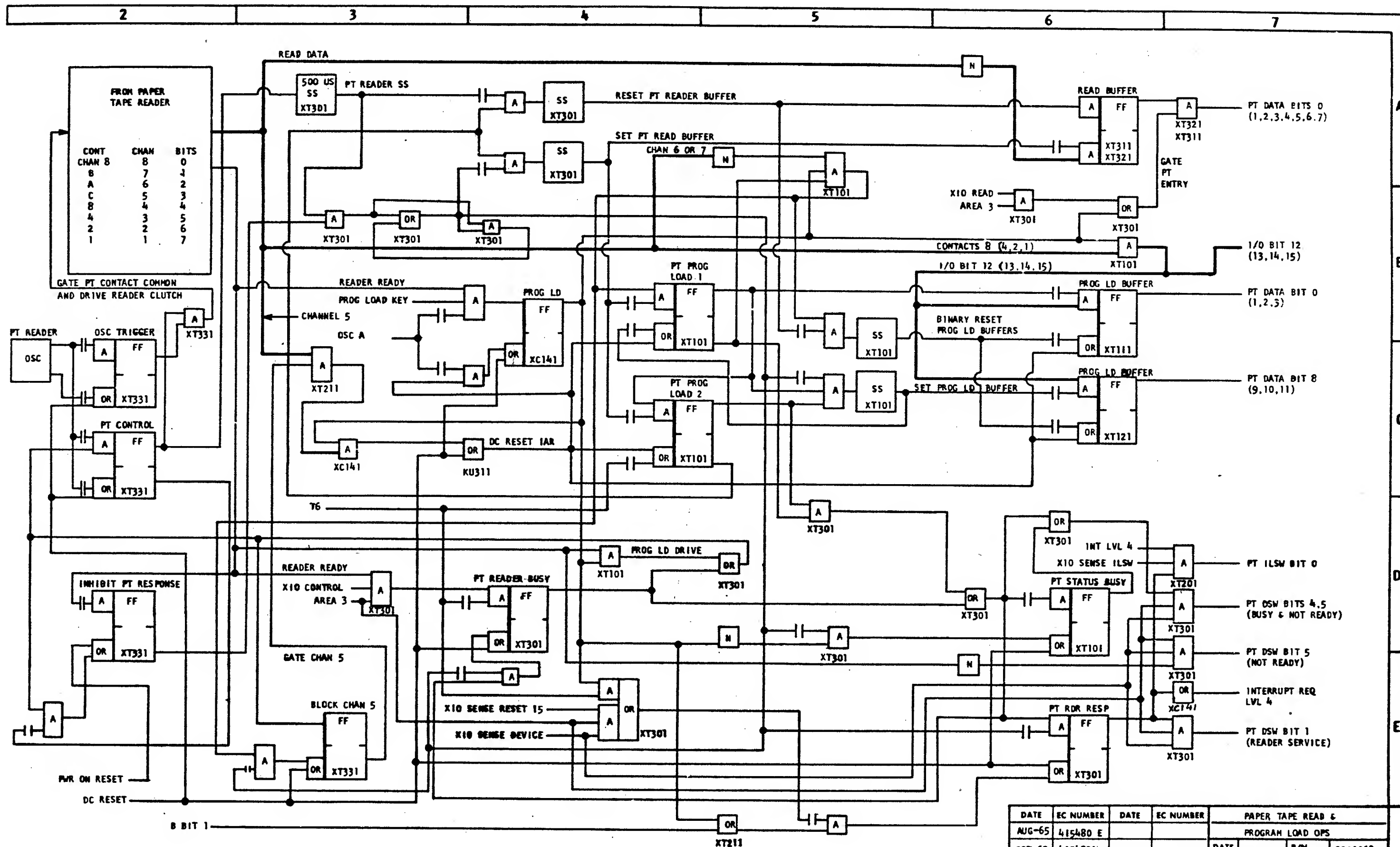




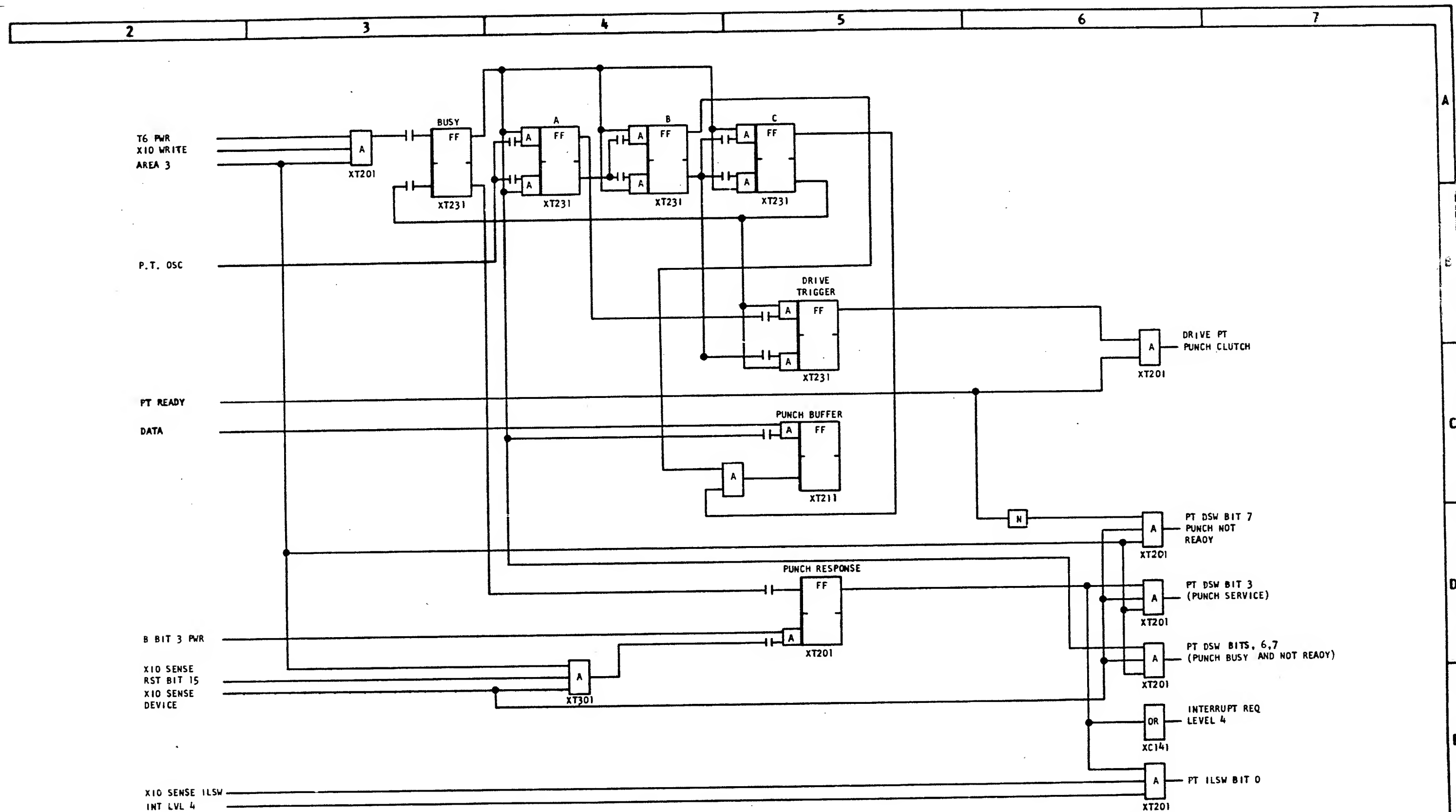
DATE	EC NUMBER	DATE	EC NUMBER	CARD READ/PUNCH			
AUG-65	415480 E			WRITE TIMING			
22APR68	419675			DATE	P/N	2201264	
					TYPE	1130	
				IBM		XR701	



DATE	EC NUMBER	DATE	EC NUMBER	CARD READ/PUNCH	
AUG-65	415480 E			CONTROL TIMING	
		DATE	3-26-65	P N	2201266
				TYPE	1130
				IBM	XR721

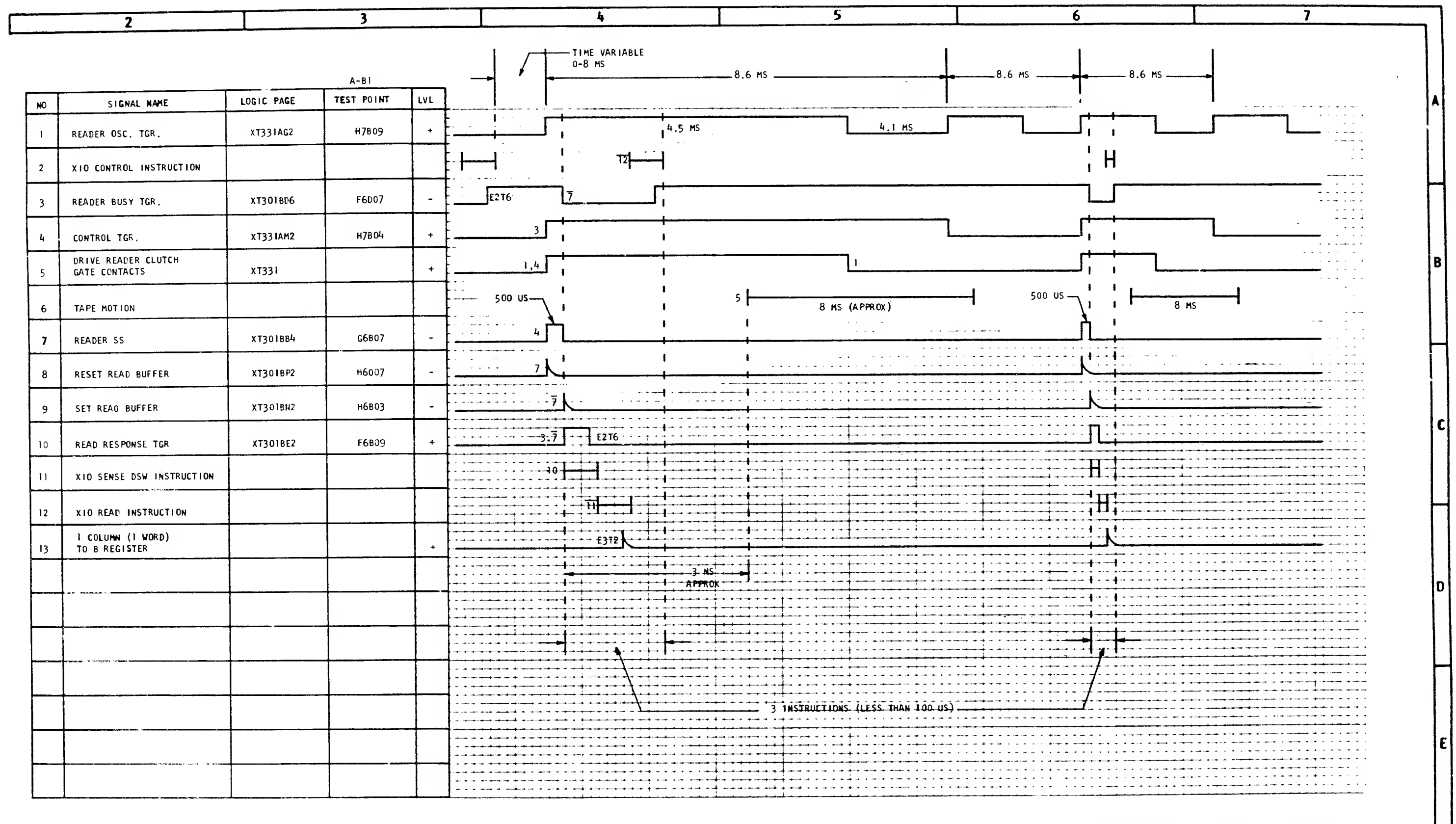


DATE	EC NUMBER	DATE	EC NUMBER	PAPER TAPE READ & PROGRAM LOAD OPS	
AUG-65	415480 E			DATE	P/N 2201268
OCT 65	415483A			TYPE	1130
JAN 66	415726			IBM XT501	
FEB 66	419603				
22APR68	419675				

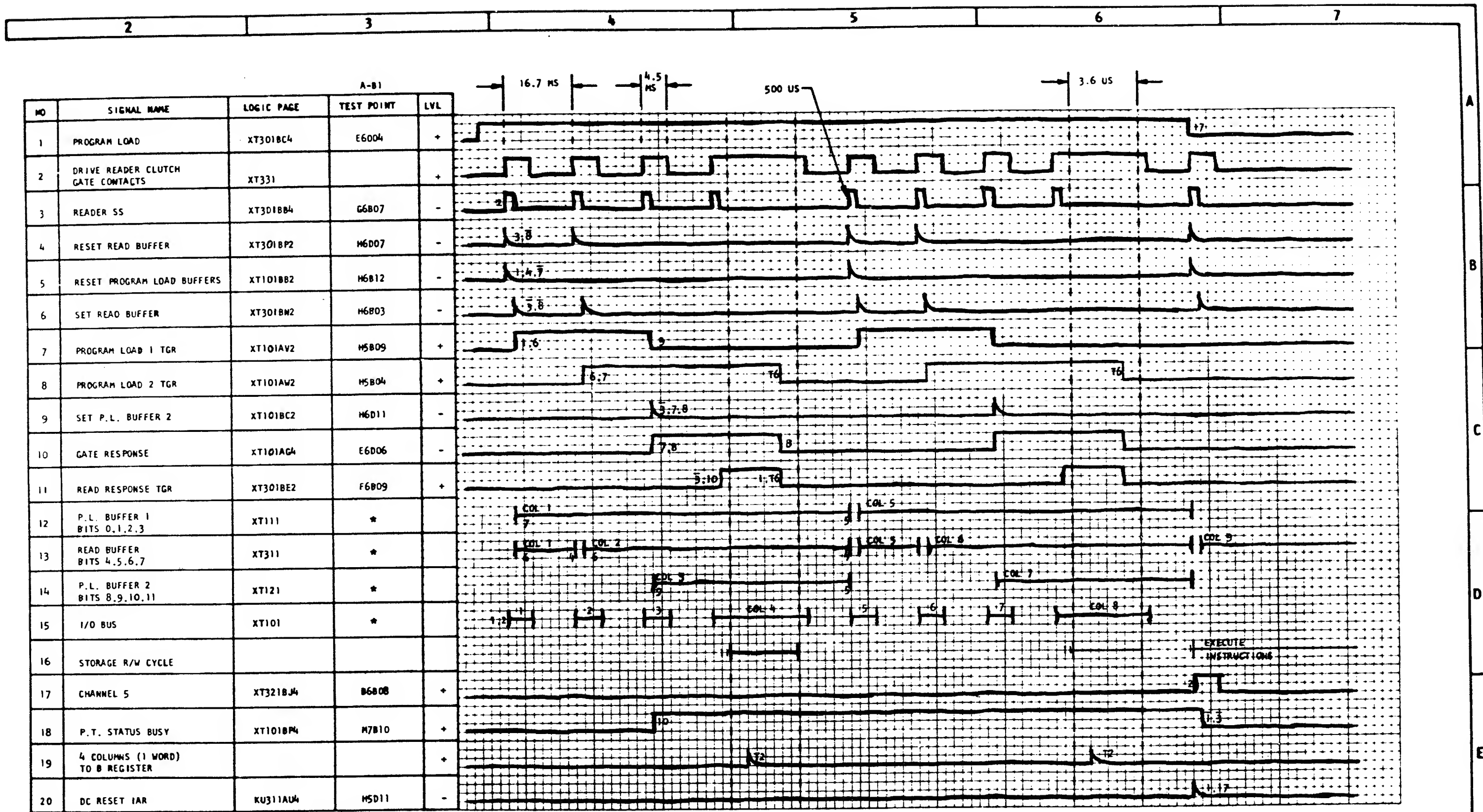


NOTE: DC RESET IS ON ALL FLIP-FLOPS

DATE	EC NUMBER	DATE	EC NUMBER	PAPER TAPE			
AUG 65	415480E			WRITE OP			
OCT 65	415483A			DATE		P/N	2201269
NOV 65	415494A					TYPE	
JAN 66	415726			IBM		XT511	
22APR68	419675						

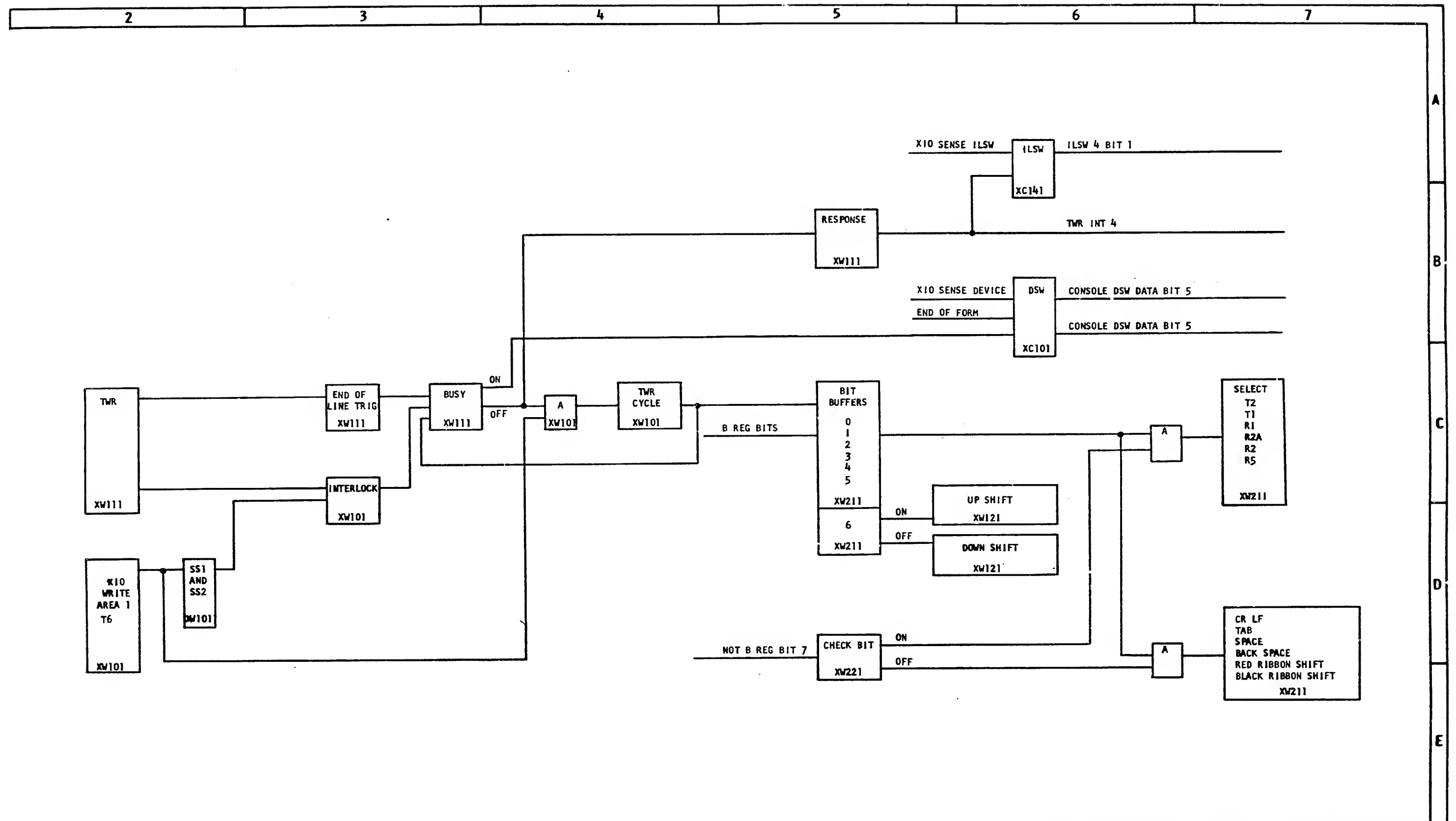


DATE	EC NUMBER	DATE	EC NUMBER	PAPER TAPE			
JAN 66	415726			READ TIMING			
22APR68	419675			DATE		P/N	2201270
						TYPE	1130
				IBM		XT701	

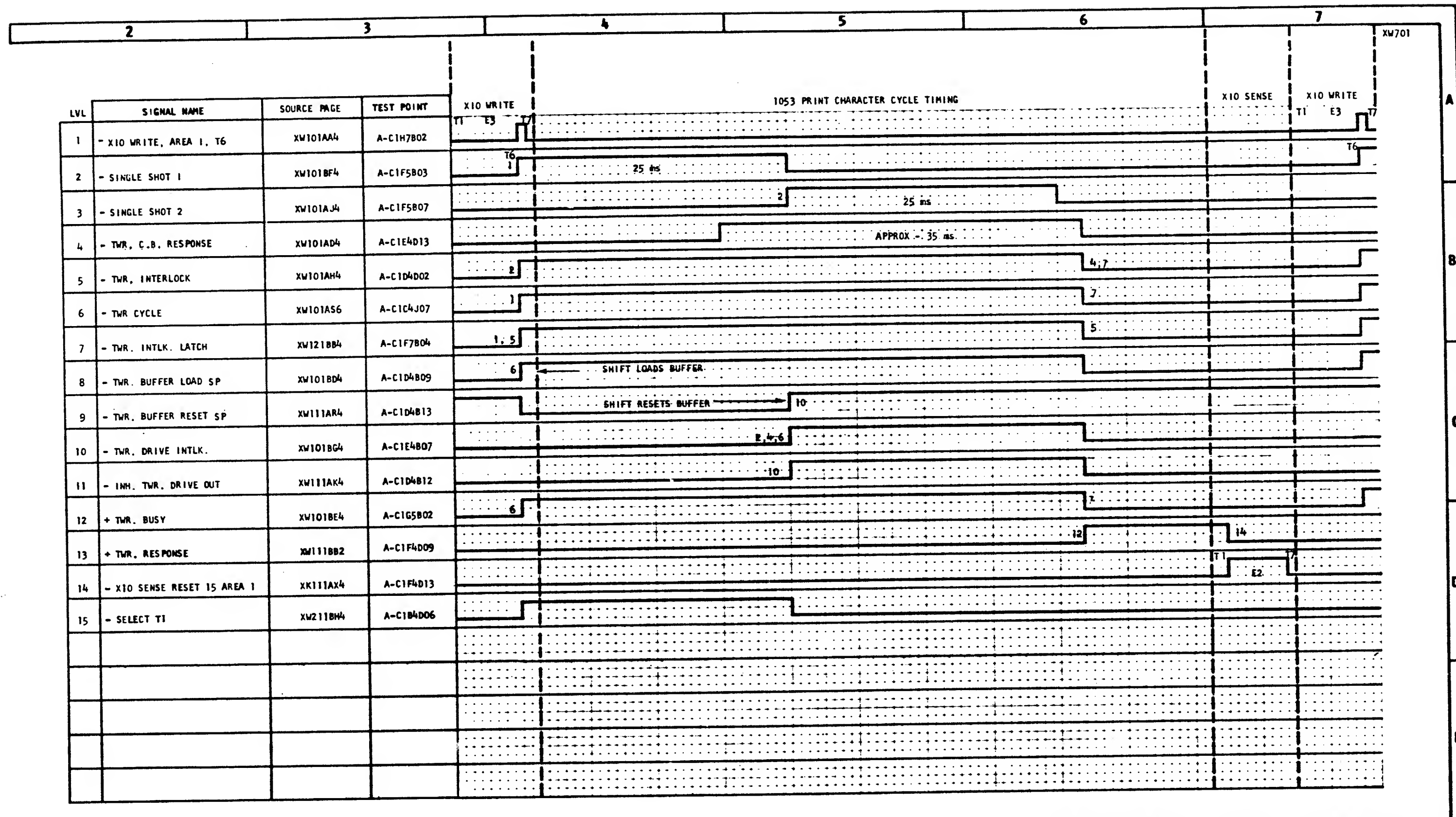


* COLUMNS 1,2,3 AND 4 CONTAIN THE 16 BITS OF THE NEXT TO LAST WORD.
COLUMNS 5,6,7 AND 8 CONTAIN THE 16 BITS OF THE LAST WORD.
COLUMN 9 CONTAINS THE CHANNEL 5 PUNCH.

DATE	SC NUMBER	DATE	SC NUMBER	PAPER TAPE			
JAN 66	415726			PROGRAM LOAD TIMING			
				DATE		P/M	2201272
						TYPE	1130
				IBM			XT721



DATE	EC NUMBER	DATE	EC NUMBER	CONSOLE PRINTER UNIT			
AUG-65	415480 E			DATA AND CONTROL DIAGRAM			
22APR68	419675			DATE		P/N	2201273
						TYPE	1130
				IBM		XW401	



DATE	EC NUMBER	DATE	EC NUMBER	CONSOLE PRINTER WRITE			
AUG-65	415480 E			AND CONTROL TIMING			
				DATE	P/M	2201275	
					TYPE	1130	
				IBM		XW701	